

Archvied in Dspace@nitr

<http://dspace.nitrkl.ac.in/dspace>

Study and Implementation of Watermarking Algorithms

A Project Report

Submitted in Partial Fulfillment of the
Requirement for the degree of

M.Tech by Research

IN

Electronics and Communication Engineering

by

Alekhika Mohanty

Under The Guidance of Dr.Ganapati Panda



Department of ECE

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA- 769008, INDIA

April 2006



NATIONAL INSTITUTE OF
TECHNOLOGY
ROURKELA, 769008
INDIA

Certificate

This is to certify that the work in this Thesis Report entitled **“STUDY AND IMPLEMENTATION OF WATERMARKING ALGORITHMS”** by **Alekhika Mohanty** has been carried out under my supervision in partial fulfillment of the requirements for the degree of **Master of Technology (Research)** in **Electronics and Communication Engineering** during session **2004-2006** in the Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela, and this work has not been submitted elsewhere for a degree.

Place: Rourkela

Date:

Dr. Ganapati Panda,
FNAE, FNASc .
Prof. ECE Department.

Acknowledgement

I would like to profusely thank my guide Prof. Ganapati Panda NIT Rourkela for his timely advice and encouragement throughout my project work. I would also like to acknowledge Prof. B. Majhi Rourkela for reviewing my work from time to time. A special note of gratitude goes to Dr. Adiga of TCS Bangalore , Dr.S.S. Meher and Dr. K.K Mohapatra of NIT for the support they have extended to this work. I would also like to thank all who helped me during my project work.

Alekhika Mohanty

<u>TABLE OF CONTENTS</u>	<u>PAGE</u>
ACKNOWLEDGMENTS	
CONTENTS	i
LIST OF TABLES	vii
LIST OF FIGURES	viii
Appendix-I	I
ABSTRACT	1
CHAPTER-1	
INTRODUCTION	
1.1 Introduction	2
1.1.1 History of Watermark	3
1.2 Watermark	4
1.2.1 Watermark Insertion	5
1.2.1.1 Watermark Generation	5
1.2.1.2 Encoding Process	5
1.2.2 Watermark Extraction	5
1.2.2.1 Decoding Process	6
1.2.2.2 Comparison Process	6
1.3 Comparison of Watermarking System With Communication System	6
1.4 Practical Challenges of Watermarking	9
1.4.1 Capacity of Watermarking Techniques	10
1.4.2 Imperceptibility	11
1.4.3 Robustness	11
1.4.3.1 Secure Watermarking	11
1.4.3.2 Robust Watermarking	12
1.4.3.3 Semi-fragile Watermarking	12
1.4.3.4 Fragile Watermarking	13
1.5 Watermarking Applications	13
1.6 Contribution of the report and thesis Organization	15

CHAPTER-2	<u>PAGE</u>
WATERMARKING ATTACKS	
2.1 Introduction	16
2.2 Classification of Attacks	16
2.2.1 On the Basis of Intent	16
2.2.1.1 Malicious Attacks	17
2.2.1.2 Non-Malicious Attack	18
2.2.2 On the Basis of Attacking Target	18
2.2.2.1 Attacks Applied to the Embedded Watermark, Type I Attacks	20
2.2.2.1 Attacks Applied to the Watermarking System, Type II Attacks	20
2.3 Measuring Attack Strength	20
2.4 Some Real Life Scenarios	22
2.4.1 Removal Attacks	22
2.4.2 Geometric Attacks	22
2.4.3 Cryptographic Attacks	23
2.4.4 Protocol Attacks	23
2.4.5 Estimation Based Attacks	24
2.4.5.1 Estimate of the Original Data	24
2.4.5.2 Remodulation Attacks	25
2.4.5.3 Copy Attack	28
2.4.5.4 Synchronization Removal	28
2.4.5.5 Benchmark Including Estimation Based Attacks	28
2.5 Counter Measure against Estimation Based Attacks	30
2.5.1 Power Spectrum Condition	32
2.5.2 Noise Visibility Function	32
2.6 Conclusion	33
CHAPTER-3	
LITERATURE REVIEW	
3.1 Introduction	34
3.2 Algorithms in Spatial Domain	34
3.3 Advantages of Frequency Based Methods	42

CHAPTER-3	<u>PAGE</u>
3.4 Algorithms in Frequency Domain	42
3.5 Advantages of Compressed Domain	51
3.6 Algorithms in Compressed Domain	52
3.7 Conclusion	58
CHAPTER-4	
REVERSIBLE WATERMARKING IN THE VLC DOMAIN OF MPEG-2	
4.1 Insight into Reversible Watermarking	59
4.2 Requirement of Online Application	63
4.3 Insight into MPEG-2 Compression	64
4.3.1 Introduction	64
4.3.2 Video Fundamentals	65
4.3.4 Bit Rate Reduction Principle	66
4.3.4.1 Spatial and Temporal Redundancy	66
4.3.4.2 Psycho visual Redundancy	66
4.3.5 Intra-frame DCT Coding	66
4.3.6 Quantization	67
4.3.7 Coding	68
4.3.8 Motion-compensated Inter-frame Prediction	75
4.3.9 MPEG-2 Details	76
4.3.9.1 Codec Structure	76
4.3.9.2 Picture Types	77
4.3.9.3 Buffer Control	79
4.4 The Proposed Algorithm	79
4.4.1 Compression	80
4.4.2 Embedding/ Detecting	83
4.4.3 Reversibility	85
4.5 Results	86
4.6 Conclusion	93

CHAPTER- 5	<u>PAGE</u>
ENERGY CLUSTER BASED WATERMARKING SCHEME USING GA	
5.1 Insight into GA	94
5.5.1 History	94
5.1.2 Chromosome	94
5.1.3 Reproduction	94
5.1.4 Encoding of a Chromosome	96
5.1.5 Crossover	96
5.1.6 Mutation	97
5.1.7 Crossover and Mutation Probability	97
5.1.7.1 Crossover Probability	97
5.1.7.2 Mutation Probability	98
5.1.8 Other Parameters	98
5.1.9 Outline of the Basic Genetic Algorithm	98
5.1.10 Genetic Algorithm Progress	99
5.1.11 Chromosome Selection Methods	100
5.1.11.1 Roulette Wheel Selection	100
5.1.11.2 Rank Selection	100
5.1.11.3 Steady-State Selection	101
5.1.11.4 Elitism	102
5.1.12 Encoding Selection Methods	102
5.1.12.1 Binary Encoding	102
5.1.12.2 Permutation Encoding	103
5.1.12.3 Value Encoding	103
5.1.12.4 Tree Encoding	104
5.1.13 Crossover and Mutation for Different Kinds of Encoding	105
5.1.13.1 Binary Encoding	105
5.1.13.2 Permutation encoding	106
5.1.13.3 Value Encoding	107

CHAPTER-5	<u>PAGE</u>
5.1.13.4 Tree Encoding	107
5.1.14 Parameters of GA	107
5.1.15 Applications of GA	108
5.2 Requirement of Watermarking Algorithm to Prove Authenticity	109
5.3 Issues Addressed in the Algorithm	109
5.4 Proposed Algorithm	110
5.4.1 Watermark Embedding Algorithm	110
5.4.1.1 Training Phase	110
5.4.1.2 The GA in Use	111
5.4.1.3 Embedding Phase	112
5.4.2 Watermark Detection Algorithm	112
5.5 Simulation Results	112
5.6 Conclusion and Scope for future Work	115
CHAPTER-6	
CONCLUSION	
6.1 Conclusion	116
BIBLIOGRAPHY	117

Appendix-I

Papers Published

<u>CONFERENCE</u>	<u>TITLE OF PAPER</u>	<u>PAGE</u>
NCSC 2006	A Novel Image Compression Scheme in the VLC Domain	I
SCT 2006	Energy Cluster-Based Watermarking Using GA	IV

LIST OF TABLES

<u>TABLE NO.</u>	<u>TITLE OF TABLE</u>	<u>PAGE</u>
1.1	Comparisons between Communication System and Watermarking System	9
3.1	Example of lc-VLC in Table B.14 of the MPEG-2 Standard	55
4.1	Extract from the MPEG-2 DCT Coefficient VLC Table	70
4.2	Table B.15 – DCT Coefficients Table	71
4.3	(Table B16 ISO/ISE 1996) – Encoding of Run and Level Following an Escape Code	75
4.5	PSNR, MSE and Bit Representation at Different Stages of the Algorithm	93
5.1	Binary string representation of encoded chromosome	96
5.2	Representation of Crossover	96
5.3	Representation of Mutation	97
5.4	Representation of Chromosome with binary encoding	102
5.5	Representation of Chromosome with Permutation Encoding	103
5.6	Representation of Chromosome with value encoding	103
5.7	Representation of Chromosome with tree encoding	104

LIST OF FIGURES

<u>FIGURE NO.</u>	<u>TITLE OF FIGURE</u>	<u>PAGE</u>
1.1	Watermark Encoder (Embedder)	7
1.2	Simple decoding Process	7
1.3	Blind Decoding Process	7
1.4	Comparing Process	8
1.5	Overall Picture of Data hiding System	8
1.6	Primary Requirements of Watermarking Algorithms	8
2.1	General Classification of Watermark Attacks on the basis of intent	19
2.2	General Classification of Watermark Attacks on the basis of attacking target	19
2.3	Distortion applied to still pictures by StirMark after bending and randomization	26
2.4	A Perceptual Remodulation Attack	26
2.5	A Copy Attack	27
2.6	A Synchronization Removal Attack	27
2.7 a)	Original Lena Image	29
2.7 b)	Watermarked work Detector output 94.6641	29
2.7 c)	After StirMark Attack Detector output 1.7644	29
2.8 a)	The Data Hider Strategy Exploiting the Texture Masking Function of HVS	31
2.8 b)	The Attacker Strategy Using Denoising and Perceptual Remodulation	31

<u>FIGURE NO.</u>	<u>TITLE OF FIGURE</u>	<u>PAGE</u>
3.1	Example of the LSB Watermarking Process. The (x.y) pairs represent the (zero run, level) pairs used in the MPEG VLC Encoding	57
3.2	Example of the Relabelling Resistant Watermarking Method	57
4.1	A Reversible Watermarking Scheme	62
4.2	Watermark embedding/extraction in raw vs. compressed video	62
4.3	The discrete cosine Transform (DCT) Pixel value and DCT coefficient magnitude are represented by dot size	69
4.4	Matrix Representing HVS Mask	69
4.5	Zigzag Scan	70
4.6 a)	Motion Compensated DCT coder	78
4.6 b)	Motion Compensated DCT decoder	78
4.7	Conversion to Frames and fields	82
4.8	Compression scheme	82
4.9	64X64 Watermark image	87
4.10	Original Lena	87
4.11	PSNR1 = 36.9859(For MPEG-2 Compressed)	87
4.12	PSNR2 = 36.9859(Compressed Using Proposed Algorithm)	87
4.13	Original Baboon	87
4.14	PSNR2 = 31.8904(MPEG-2 Compressed)	87
4.15	PSNR2 = 31.8904(Compressed using Proposed Algorithm)	88

<u>FIGURE NO.</u>	<u>TITLE OF FIGURE</u>	<u>PAGE</u>
4.16	Original Barbara	88
4.17	PSNR1 = 34.9343(For MPEG-2 Compressed)	88
4.18	PSNR2 = 34.9343(Compressed using Proposed Algorithm)	88
4.19	Original Goldhill	88
4.20	PSNR1 = 34.7470(For MPEG-2 Compressed)	88
4.21	PSNR2 = 34.7470(Compressed using Proposed Algorithm)	89
4.22	Original Peppers	89
4.23	PSNR2 = 34.9343(For MPEG-2 Compressed)	89
4.24	PSNR2 = 34.9343(For MPEG-2 Compressed)	89
4.25	Absolute Difference between Original Lena & Recovered Lena	89
4.26	Absolute Difference between Original Baboon & Recovered Baboon	89
4.27	Absolute Difference between Original Barbara & Recovered Barbara	90
4.28	Absolute Difference between Original Goldhill & Recovered Goldhill	90
4.29	Absolute Difference between Original Peppers & Recovered Peppers	90
4.30	Absolute Difference between Original Lena & Reconstructed Lena	90
4.31	Absolute Difference between Original Baboon & Reconstructed Baboon	90

<u>FIGURE NO.</u>	<u>TITLE OF FIGURE</u>	<u>PAGE</u>
4.32	Absolute Difference between Original Goldhill & Reconstructed Goldhill	90
4.33	Absolute Difference between Original Barbara & Reconstructed Barbara	91
4.34	Absolute Difference between Original Peppers & Reconstructed Peppers	91
4.34	Number of Bits in Raw, MPEG-2 Compressed Data and Proposed Algorithm Compressed Data	92
5.1	Representation o Roulette Wheel Selection	100
5.2 a)	Situation before Ranking (graph of fitness)	101
5.2 b)	Situation after Ranking (graph of order numbers)	101
5.3 a)	Single Point Crossover	105
5.3 b)	Two Point Crossover	105
5.3 c)	Uniform Crossover	106
5.3 d)	Arithmetic Crossover	106
5.4	Bit Inversion Mutation	106
5.5	Tree Encoding Crossover	107
5.6	512X512 Original Lena Image	113
5.7	64x64 WATERMARK IMAGE	113
5.8	The Watermarked image generated by Proposed algorithm PSNR 39.671	113
5.9	The Watermarked image generated by Embedding into random pixels PSNR 37.671	113
5.10 a)	Watermark recovered from proposed algorithm NC = 0.9431	114
5.10 b)	For the proposed algorithm Watermark recovered after mean filtering of the watermarked image NC = 0.5388	114

<u>FIGURE NO.</u>	<u>TITLE OF FIGURE</u>	<u>PAGE</u>
5.10 c)	For the proposed algorithm Watermark recovered after mean filtering of the watermarked image NC= 0.5372	114
5.11 a)	For the proposed algorithm Watermark recovered after mean filtering of the watermarked image.	114
Figure 5.11 b)	For the proposed algorithm Watermark recovered after mean filtering of the watermarked image .Watermark cannot be detected	114

**“IF I HAVE SEEN FURTHER
IT IS BY STANDING UP ON THE SHOULDERS OF GIANTS”**

Sir Issac Newton

Dedicated
To
Mama, Baba

Abstract

Watermarking is the process of embedding data called a watermark into a multimedia object such that watermark can be detected or extracted later to make an assertion about the object. The object may be an audio, image or video. A copy of a digital image is identical to the original. This has in many instances, led to the use of digital content with malicious intent. One way to protect multimedia data against illegal recording and retransmission is to embed a signal, called digital signature or copyright label or watermark that authenticates the owner of the data. Data hiding, schemes to embed secondary data in digital media, have made considerable progress in recent years and attracted attention from both academia and industry. Techniques have been proposed for a variety of applications, including ownership protection, authentication and access control. Imperceptibility, robustness against moderate processing such as compression, and the ability to hide many bits are the basic but rather conflicting requirements for many data hiding applications. A simple example of digital watermark would be a visible “seal” placed over an image to identify the copyright. However the watermark might contain additional information including the identity of the purchaser of a particular copy of the object.

In this thesis issues involving digital watermarking and its applications have been addressed. In addition, a few other important problems encountered in practice, such as the requirement for **reversibility** of the original data after the watermark has been extracted, have been discussed here. Various issues of watermarking are studied with new principles and techniques being proposed. Extensive survey of current watermarking literatures has been done. Two watermarking algorithms have been proposed in this work. One is a reversible watermarking scheme in the VLC domain of MPEG-2 data. Other is an evolutionary watermarking scheme using genetic algorithm. The two algorithms appear under the following headings in this thesis.

- A Reversible Watermarking Algorithm in the VLC domain of MPEG-2 Data
- An Energy Cluster Based Watermarking Algorithm Using Genetic Algorithm

CHAPTER 1

INTRODUCTION

OBJECTIVE: Chapter 1 provides insight into Watermarking, focuses on its practical challenges and applications.

CHAPTER ORGANISATION:

1.1 Introduction

1.2 Watermark

1.3 Comparison of Watermarking System with Communication System

1.4 Practical Challenges of Watermarking

1.5 Watermarking Applications

1.6 Contribution of the report and thesis organization

1.1 Introduction

This is the digital information revolution era. It has heralded connectivity, i.e. connectivity over the Internet and connectivity through the wireless network. Innovative devices such as digital camera and camcorder, high quality scanners and printers, digital voice recorder, MP3 player and PDA, have reached consumers worldwide to create, manipulate, and enjoy the multimedia data. The development of high speed computer networks and that of internet, in particular, has explored means of new business, scientific, entertainment, and social opportunities in the form of electronic publishing and advertising, real-time information delivery, product ordering, transaction processing, digital repositories and libraries, web newspapers and magazines, network video and audio, personal communication etc. The cost effectiveness of selling software, high quality art work in the form of digital images and video sequences by transmission over **World Wide Web** (www) is greatly enhanced as a consequence of technological improvement. The commercial exploitation of www is steadily being more appreciated.

The boom in the information age is not without its adverse effects though. Copying is simple with no loss of fidelity. A copy of a digital media is identical to the original. This has in many instances, led to the use of digital content with malicious intent. One way to protect multimedia data against illegal recording and retransmission is to embed a signal, called **digital signature** or **copyright label** or **watermark** that authenticates the owner of the data. With the ease of editing and perfect reproduction in digital domain, the protection of ownership and the prevention of unauthorized tampering of multimedia data (audio, image, video, and document) have become important concerns. Data hiding, schemes to embed secondary data in digital media, have made considerable progress in recent years and attracted attention from both academia and industry. Techniques have been proposed for a variety of applications, including ownership protection, authentication and access control. Imperceptibility, robustness against moderate processing such as compression, and the ability to hide many bits are the basic but rather conflicting requirements for many data hiding applications.

1.1.1 History of Watermark

The earliest reference to Watermarking in history dates back to the B.C era. The present day Watermarking has developed basically from two different streams, Cryptography meaning, “secret writing” and Steganography, which in the Greek language means, “cover writing”. **Cryptography** is the study of methods of sending messages in distinct form so that only the intended recipients can remove the disguise and read the message. The intended message to be sent is called **plain text** message and the disguised message is called **cipher text**. The process of converting a plain text to a cipher text is called **enciphering** or **encryption**, and the reverse process is called **deciphering** or **decryption**. Encryption protects contents during the transmission of the data from the sender to receiver. However, after receipt and subsequent decryption, the data is no longer protected and is in the clear. Reference to cryptography and some of its applications can be found in [1-7].

Steganography is the art of devising astute and undetectable methods of concealing the message themselves. It is therefore broader than cryptography. There is no theory for steganography. The origin of steganography is biological and physiological [8-9]. The earliest allusion to secret writing in the west appears in Homer’s Iliad [10]. Steganographic methods made their record debut a few centuries later in several tales by Herodotus, the father of history [11]. Some of them can also be found in [12]. Few other examples of steganography can be found in [12]. An important technique was the use of sympathetic inks. Ovid in his “Art of Love” suggests using milk to write invisibly. Later, chemically affected sympathetic inks were developed. This was used in World Wars 1 and 2. Ancient references to secret writing and steganography also appear in Asia. Indian literature is replete with references as well as explicit formulas for secret writing. Kautilya’s Arthashastra and LalitaVistara, and Vatsayana’s Kamasutra are few famous examples. In ancient China, military and diplomatic rulers wrote important messages on thin sheets of silk or paper. For secure transport, the sheets were rolled into balls, covered with wax, and swallowed by and placed in the rectum of messenger [13].

Watermarking techniques are particular embodiments of steganography. The use of watermarks is almost as old as paper manufacturing. Our ancients poured their half-stuff

slurry of fiber and water on to mesh molds to collect the fiber, then dispersed the slurry within deckle frames to add shape and uniformity, and finally applied great pressure to expel the water and cohere the fiber. This process hasn't changed too much in 2000 years. One by-product of this process is the **watermark** – the technique of impressing into the paper a form of image, or text derived from the negative in the mold, as the paper fibers are squeezed and dried. Paper Watermarks have been in wide use since the late Middle Ages. Their earliest use seems to have been to record the manufacturer's trademark on the product so that the authenticity could be clearly established without degrading the aesthetics and utility of the stock. In more recent times, watermarks have been used to certify the composition of paper, including the nature of the fibers used. Today most developed countries also watermark their paper, currencies and postage stamps to make forgery more difficult.

The digitization of our world has expanded the concept of watermarking to include immaterial digital impressions for use in authenticating ownership claims and protecting proprietary interests. However, in principle digital watermarks are like their paper ancestors. Watermarks of varying degree of visibility are added to presentation media as a guarantee of authenticity, quality ownership and source both as the product of paper press or discrete cosine transformations. Digital watermarking differs from digital fingerprinting which produces a metafile that describes the contents of the source file [14, 15].

1.2 Watermark

In this thesis, work has been carried out on Digital Watermarking. Throughout the rest of the report, watermarking refers to digital watermarking. To avoid the unauthorized distribution of images or other multimedia property, various solutions have been proposed. Most of them make unobservable modifications to images that can be detected afterwards [16]. Such image changes are called watermarks. Watermarking is defined as adding (embedding) a payload signal to the host signal. The payload can be **detected** or **extracted** later to make an assertion about the object i.e. the original data that may be an **image** or **audio** or **video**.

In general, any watermarking scheme (algorithm) consists of three parts: [17]

- **The watermark** (payload)
- **The encoder** (marking insertion algorithm)
- **The decoder and comparator** (verification or extraction or detection algorithm)

Each owner has a unique watermark or an owner can also put different watermarks in different objects the marking algorithm incorporates the watermark into the object. The verification algorithm authenticates the object determining both the owner and the integrity of the object.

1.2.1 Watermark insertion:

Watermark insertion involves watermark generation and encoding process

1.2.1.1 Watermark Generation:

Each owner has a unique watermark or an owner can also put different watermarks in different objects the marking algorithm incorporates the watermark into the object. The verification algorithm authenticates the object determining both the owner and the integrity of the object. The watermark can be a logo picture, sometimes a binary picture [18], sometimes a ternary picture [19]; it can be a bit stream [16] or also an encrypted bit stream etc. The encryption may be in the form of a hash function [21] or encryption using a secret key [20, 22]. The watermark generation process varies with the owner.

1.2.1.2 Encoding Process:

In the encoding process both the original data and the payload data are passed through the encoding function. The payload signal and the original host signal now together occupy space, which was previously occupied only by the host signal. For this purpose either the original data is compressed [1, 3, 4, 5 6] or redundancy in digital content is explored to make space for the payload [2].

1.2.2 Watermark Extraction:

Extraction is achieved in two steps. First the watermark or payload is extracted in the decoding process and then the authenticity is established in the comparing process.

1.2.2.1 Decoding Process:

The decoding process can be itself performed in two different ways. In one process the presence of the original unwatermarked data is required and other where blind decoding is possible. Fig.1.2 and Fig.1.3 show the two processes. A decoder function takes the test data (the test data can be a watermarked or un-watermarked and possibly corrupted) whose ownership is to be determined and recovers the payload.

1.2.2.2 Comparison Process:

The extracted payload is compared with the original payload (i.e. the payload that was initially embedded) by a comparator function and a binary output decision is generated. The comparator is basically a correlator. Depending on the comparator output it can be determined if the data is authentic or not. If the comparator output is greater than equal to a threshold then the data is authentic else it is not authentic.

Fig.1.4 illustrates the comparing function. In this process the extracted payload and the original payload are passed through a comparator. The comparator output C is the compared with a threshold and a binary output decision generated. It is 1 if there is a match i.e. $C \geq \delta$ and 0 otherwise.

A watermark is detectable or extractable to be useful [20]. Depending on the way the watermark is inserted and depending on the nature of the watermarking algorithm, the method used can involve very distinct approaches. In some watermarking schemes, a watermark can be extracted in its exact form, a procedure we call **watermark extraction**. In other cases, we can detect only whether a specific given watermarking signal is present in an image, a procedure we call **watermark detection**. It should be noted that **watermark extraction can prove ownership** whereas **watermark detection can only verify ownership** [21].

1.3 Comparison of Watermarking System with Communication System

The Watermarking System can be compared to a communication system [23]. Like the Communication System the watermarking system consists of three parts. The analogy of communication system and watermarking system is shown in table 1.1 and Fig.1.5.

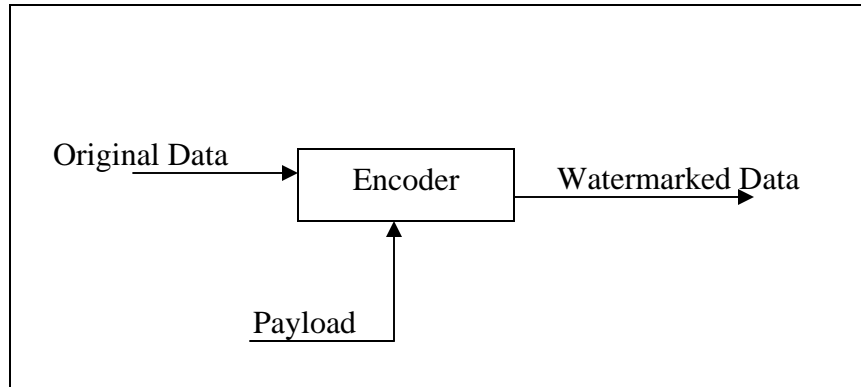


Figure 1.1 Watermark Encoder (Embedder)

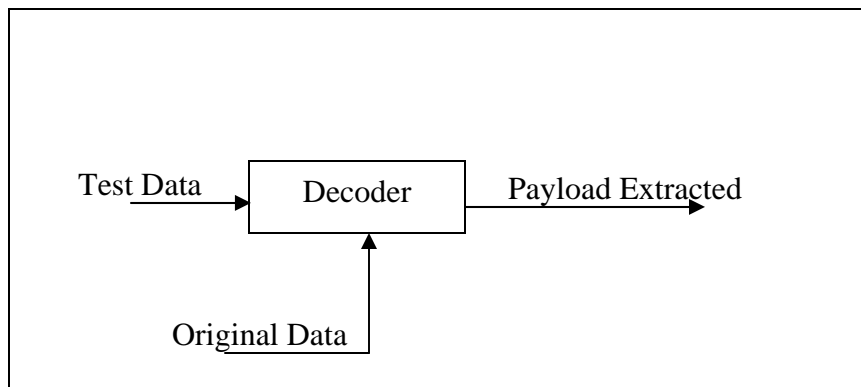


Figure 1.2 Simple decoding Process

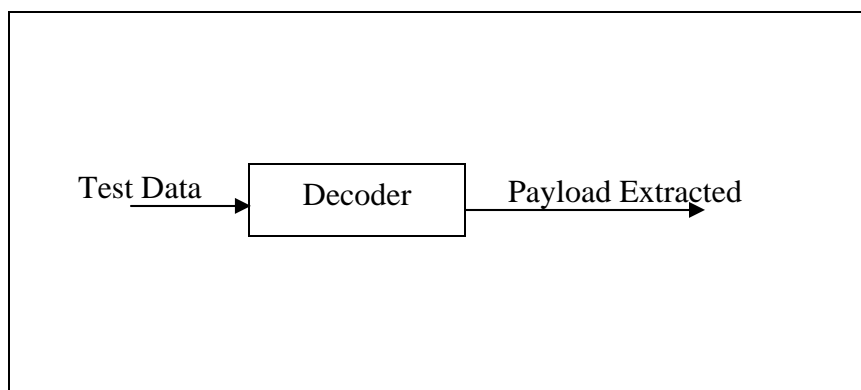


Figure 1.3 Blind Decoding Process

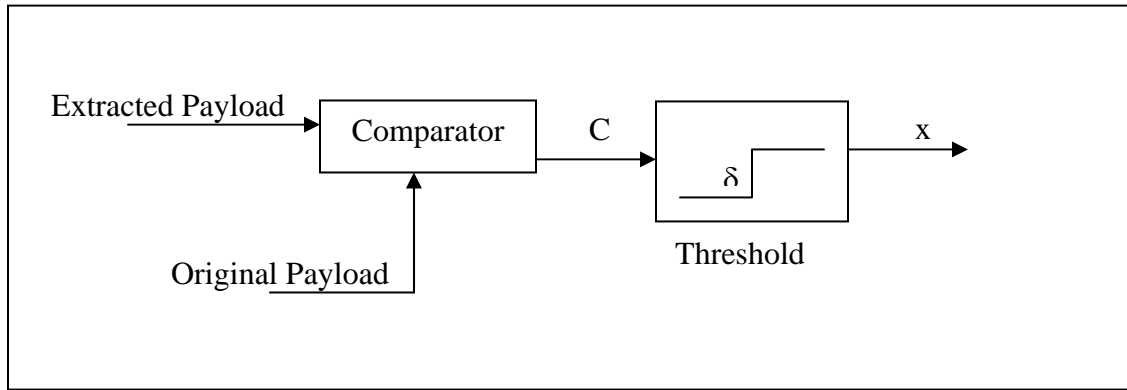


Figure 1.4 Comparing Process

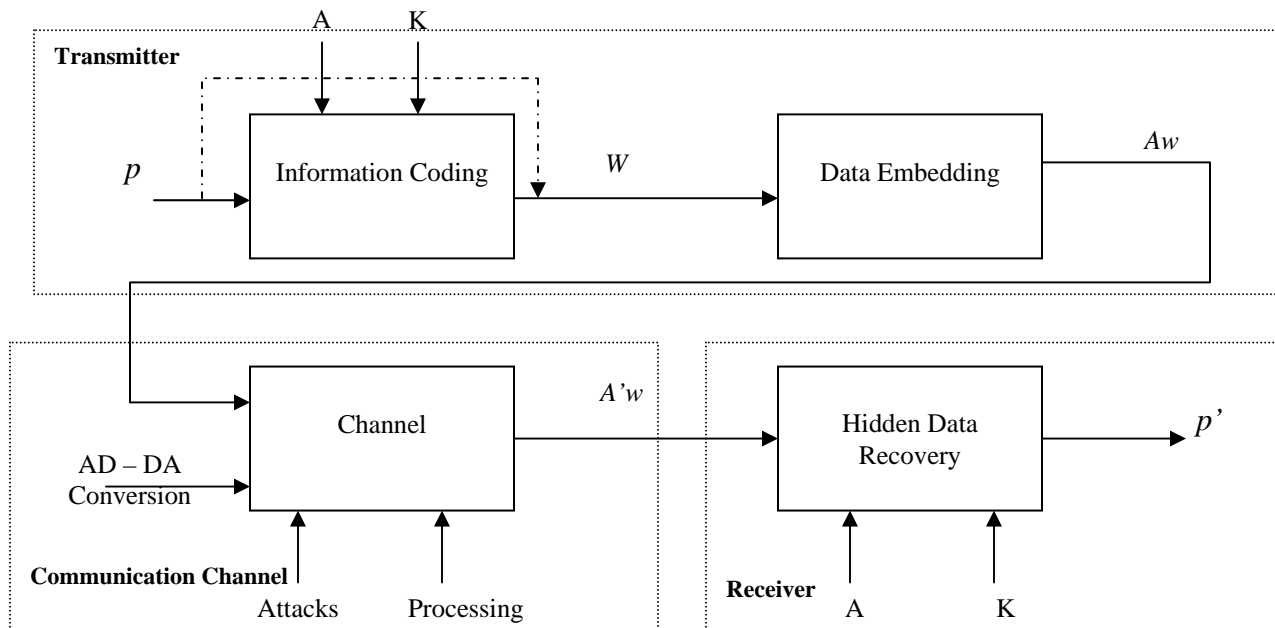


Figure 1.5 Overall Picture of Data hiding System

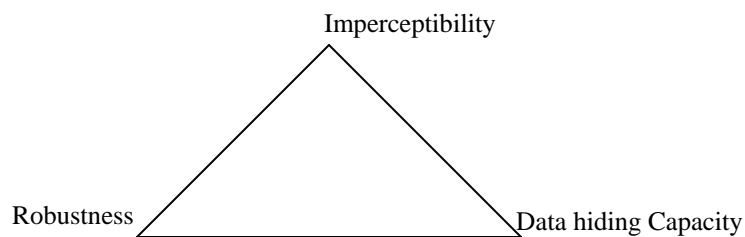


Figure 1.6 Primary Requirements of Watermarking Algorithms

Table 1.1 Comparisons between Communication System and Watermarking System

Communication System	Watermarking System
Transmitter	Embedding of the to-be-hidden information in the host signal (Embedding)
Transmission through the communication channel	Any processing applied to the host data after information concealment, along with the interaction between the concealed data and the host data. (Attacks + processing)
Receiver	Recovery of the hidden information from the host data (Decoding)

In Fig. 1.5 the host data is depicted as asset 'A'. 'A' may be an audio file, a still image, video or a combination of audio and video. Throughout the thesis though only still images have been handled. So in the rest of the work 'A' refers to still images. The information embedded is depicted as payload ' p '. Then p is transformed in a watermark signal ' w ' (optionally $p = w$). The embedding module may accept a secret key ' K ' as an additional input. Watermarked asset ' A_w ' is formed as a result of this encoding. The key ' K ' introduces some secrecy within the embedding step. Due to possible attacks ' A_w ' is transformed to ' $A'w$ '. Finally the decoder/detector recovers the hidden information from ' $A'w$ '.

1.4 Practical Challenges of Watermarking

Watermark by itself is not sufficient to prevent abuses unless a proper protection protocol is established. The exact properties that a watermarking algorithm must satisfy cannot be defined exactly without considering the particular application scenario, the algorithm has to be used in. A brief analysis of requirements of data hiding algorithms from a protocol perspective permits to decide whether a given algorithm is suitable for a certain application or not.

Each watermarking application has its own specific requirements. Most often than not these requirements have conflicting effects on each other. A good watermarking

algorithm obtains optimal tradeoff between these requirements; is not weakened/destroyed by attacks, both malicious and non-malicious; at the same time unambiguously identifies the owner. These properties can be broadly classified as primary and secondary requirements. The primary requirements include data hiding capacity, imperceptibility and robustness as shown in figure 1.6. However these three characteristics conflict with each other. Increasing fidelity of the watermarked images (i.e. increasing imperceptibility of the mark) would lower the strength of the watermark. Embedding large amount of information reduces the fidelity of the watermark. The secondary requirements include performance i.e. the speed of embedding and of detection of the watermark. These attributes though less commonly discussed are very important for many real world applications.

Each of the primary attributes has been discussed in detail below.

1.4.1 Capacity of Watermarking Techniques

Capacity is a fundamental property of any watermarking algorithm, which very often determines whether a technique can be profitably used in a given context or not. However no requirement can be set without considering the application the technique has to serve in.

Possible requirements range from some hundreds of bits in security-oriented applications, where robustness is a major concern, through several thousands of bits in applications like captioning or labeling, where the possibility of embedding a large number of bits is a primary need. For copy protection purposes, a payload of one bit is usually sufficient. According to a recent proposal for audio watermarking technology from the International Federation for the Phonographic industry,(IFPI), the minimum payload for an audio watermark should be 20 bits per second, independently of the signal level and music type[24]. However according to [25] this minimum is very ambitious and should be lowered to only a few bits per second. For the protection of intellectual properties rights, the payload that would suffice is about 60 bits [26] or 70 bits [27] per image, video-frame or audio fragment.

Capacity requirements always struggle against two other important requirements, watermark imperceptibility and watermark robustness. A higher capacity is always

obtained at the expense of either robustness or imperceptibility or both. It is therefore mandatory that a good trade-off be found depending on the application at hand.

1.4.2 Imperceptibility

The watermark should be imperceptible so as not to affect the viewing experience of the image or the quality of the audio signal. In most applications the watermarking algorithm must embed the watermark such that this does not affect the quality of the underlying host data. A watermark-embedding procedure is truly imperceptible if humans cannot distinguish the original data from the data with the inserted watermark [13]. However even the smallest modification in the host data may become apparent when the original data is compared directly with the watermarked data. Since users of watermarked data normally do not have access to the original data, they cannot perform this comparison. Therefore, it may be sufficient that the modifications in the watermarked data go unnoticed as long as the data are not compared with the original data [29].

1.4.3 Robustness

Watermark robustness accounts for the capability of the hidden data to survive host signal manipulations, including both non-malicious manipulations, which do not explicitly aim at removing the watermark or at making it unreadable, and malicious manipulations, which precisely aim at damaging the hidden information.

The exact level of robustness the hidden data must possess cannot be specified without considering a particular application. Qualitative robustness levels encompassing most of the situations encountered in practice have been discussed below.

1.4.3.1 Secure Watermarking:

In this case, mainly dealing with copyright protection, ownership verification or other security-oriented applications, the watermark must survive both non-malicious as well as malicious manipulations. In secure watermarking, the loss of the hidden data should be obtainable only at the expense of a significant degradation of the quality of the host signal. When considering malicious manipulations it has to be assumed that attackers know the watermarking algorithm [30] and thereby they can conceive ad-hoc watermark

removal strategies. The security must lie on the choice of a key. The watermarking algorithm is truly secure if the knowing the exact algorithms for embedding and extracting the watermark does not help unauthorized party to detect the presence of the watermark. As to non-malicious manipulations, they include a huge variety of digital and analog processing tools, including lossy compression [31], linear and non-linear filtering cropping, editing, scaling, D/A and A/D conversions, analog duplications, noise addition and many others that apply only to particular type of media. Thus, in the image case, we must consider zooming and shrinking, rotation, contrast, enhancement histogram manipulations, row/column removal or exchange, in the case of video we must take into account frame removal, frame exchange, temporal filtering, temporal resampling, finally robustness of an audio watermark, may imply robustness against echo addition, multirate processing, reverb, wow-and -flutter, and pitch scaling. It is though important to point out that even the most secure system does not need to perfect the contrary, it is only needed that a high enough degree of security is reached. In the other words, watermark breaking does not need to be impossible (which probably will never be the case), but only difficult enough.

1.4.3.2 Robust Watermarking:

In this case it is required that the watermark be resistant only against non-malicious manipulations. Robust watermarking is less demanding than secure watermarking. Application fields in robust watermarking include all the situations in which it is unlikely that someone purposely manipulates the host data with the intention to remove the watermark. The application scenario is such that the normal use of data comprises of several kinds of manipulations, which must not damage the hidden data. Even in copyright protection applications, the adoption of robust watermarking instead of secure watermarking may be allowed due to the use of a copyright protection protocol in which all the involved actors are not interested in removing the watermark.

1.4.3.3 Semi-fragile Watermarking:

Watermark is semi-fragile if it survives a limited well specified, set of manipulations, leaving the quality of the host document virtually intact. In some applications robustness

is not a major requirement, mainly because the host signal is not intended to undergo any manipulations, but a very limited number of minor modifications such as moderate lossy compressions, or quality enhancement. This is the case of data labeling for improved actual retrieval, in which the hidden data is only needed to retrieve the host data from archive, and thereby it can be discarded once the data has been correctly assessed. Usually data is archived in compressed format, and that the watermark is embedded prior to compression. In this case the watermark needs to be robust against lossy coding.

1.4.3.4 Fragile Watermarking:

A watermark is said to be fragile if the information hidden within the host data is lost or irremediably altered as soon as any modification is applied to the host signal. Such a loss of information may be global, i.e. no part of watermarking can be recovered, or local i.e. only part of the watermark is damaged. The main application of fragile watermarking is data authentication, where watermark loss or alteration is taken as evidence that the data has been tampered with. The recovery of the information content within the data demonstrates authentic un-tampered data.

Robustness against signal distortion is better achieved if the watermark is placed in perceptually significant parts of the signal. This is particularly evident in the case of lossy compression algorithms, which operate by discarding perceptually insignificant data. Watermarks hidden within perceptually insignificant data are likely not to survive compression. Achieving watermark robustness, and, to a major extent, watermark security is one of the main challenges watermarking researches are facing with. Nevertheless its importance has sometimes been over estimated at the expense of other very important issues such as watermark capacity and protocol level analysis.

1.5 Watermarking Applications

Recently there has been an explosion in the use and distribution of digital multimedia data. Personal computers with (broadband) internet connections have become more and more common, and have made the distribution of multimedia data and applications much easier and faster. Electronic commerce applications and online services are rapidly being developed. Even the analog home audio and video equipment are rapidly being replaced

by digital successors. As a result, digital mass recording devices for multimedia data have entered today's consumer market. Digital data has many advantages over analog data. However, it also opens the possibility of unrestricted duplication and manipulation of copyright material.

To prevent the unauthorized access or manipulation of digital multimedia data, two complementary techniques can be used, namely encryption and watermarking [32]. Encryption techniques can be used to protect digital data during the transmission from the sender to the receiver [33]. However, after the receiver has received and decrypted the data, the data is identical to the original data and no longer protected. Watermarking techniques complement encryption by embedding a secret imperceptible signal, a watermark, directly into the original data in such a way that it always remains present. Such a watermark is used for the following purposes [34]:

1. Copyright protection: A watermark is used to carry copyright information as a proof in case of a copyright or ownership dispute.
2. Fingerprinting: Unique information, directly coupled to user identification, is embedded in the data as a watermark. In case of copyright violation, this watermark can be used to trace the source of illegal copies.
3. Copy protection: A watermark is used to carry information prohibiting copying of protected data on compliant hardware.
4. Broadcast monitoring: A watermark is embedded into data, for example, commercials or copyrighted materials [35], to allow automatic monitoring of the data in the broadcasting channels. The results of this monitoring can be used for royalty or copyright protection purposes.

Digital watermarking can also be in other applications not dealing with copy or copyright protection:

1. Indexing: Indexing of video mail, where comments can be embedded in the video content: indexing of movies and news items where markers and comments can be inserted that can be used by search engines.
2. Medical application: Embedding the date and the patient's name in the medical images could be useful safety measure.

3. Data embedding: Watermarking techniques can be used to embed messages in the data. The data can be secret or private, but it can also be public. An example of the latter is Digimarc's Smart Images [36].
4. Error detection /Tamper proofing: In [37], the authors presented an error detection scheme in video coding using a fragile watermark. The authors show that this proposed scheme performs significantly better than a syntax-based error detection scheme. Similar approaches are also presented in [39, 40].
5. Compression: The authors in [38] use watermarking techniques to improve the compression rate of color images. In this scheme, the color information of the image is embedded as a watermark into the luminance data to reduce the data storage requirements.

1.6 Contribution of the report and thesis organization

This thesis addresses the issues regarding Digital Watermarking and its applications. In addition, a few other important problems encountered in practice, such as the requirement for **reversibility** of the original data after the watermark has been extracted, have been discussed here. The works included in this thesis intend to contribute toward the understanding of digital data hiding. Various issues of watermarking are studied with new principles and techniques being proposed.

Two watermarking algorithms have been proposed in this work. One is a reversible watermarking scheme in the VLC domain of MPEG-2 data. Other is an evolutionary watermarking scheme using genetic algorithm.

Chapter 1 is an introduction to Watermarking. Chapter 2 deals with watermarking attacks. Literature survey has been presented in chapter 3. The proposed reversible watermarking algorithms, the evolutionary watermarking algorithm, have been presented in chapter 4 and chapter 5 respectively. Chapter 6 is the concluding chapter.

CHAPTER 2

WATERMARKING ATTACKS

OBJECTIVE: To define and enumerate the different attacking algorithms that can affect the hidden data because of manipulations undergone

CHAPTER ORGANISATION:

- 2.1 Introduction
- 2.2 Classification of Attacks
- 2.3 Measuring Attack Strength
- 2.4 Some Real Life Scenarios
- 2.5 Counter Measure against Estimation Based Attacks
- 2.6 Conclusion

2.1 Introduction

To win each campaign, a general needs to know about both his opponent's as well as his own troops. Attacks aim at weakening the watermarking algorithm. The purpose of any watermark-embedding algorithm is to provide some degree of security and the purpose of any attack is to negate that purpose. Hence the compilation of a report on watermarking is incomplete without a mention of watermarking attacks.

Study of watermarking algorithm enable to

- Identify weakness of the watermarking algorithm
- Propose improvement of the watermarking algorithm
- Study effects of current technology on watermark

In watermarking terminology, an attack is any processing that may impair detection of the watermark or communication of the information conveyed by the watermark. The processed watermarked data is then called attacked data.

Watermarking is treated as a communications problem [23], in which the owner attempts to communicate over a hostile channel, where the non-intentional and the intentional attacks form the channel. The owner tries to communicate as much watermark information as possible while maintaining a sufficient high data quality. Contrary, an attacker tries to impair watermark communication while impairing the data quality as little as possible. Therefore, digital watermarking scenarios can be considered as a game between the owner and the attacker. Continuing with the analogy of watermarking as a communications system, some researchers have chosen to work on modeling and resisting attacks on the watermark. They work on the philosophy that the more specific the information known about the possible attacks, the better we can design systems to resist it.

The detailed classification of attacks is listed below:

2.2 Classification of Attacks:

2.2.1 On the Basis of Intent:

These attacks can be broadly classified as non-malicious [54] (unintentional) such as compression of a legally obtained, watermarked image or video file and malicious (intentional) such as an attempt by a multimedia pirate to destroy the embedded information and prevent tracing of illegal copies of watermarked digital video. These are

shown in Fig. 2.1. Watermarking systems utilized in copy protection or data authentication schemes are especially susceptible to malicious attacks. Non-malicious attacks usually come from common signal processing operations done by legitimate users of the watermarked materials.

2.2.1.1 Malicious Attack: An attack is said to be malicious if its main goal is to remove or make the watermark unrecoverable.

Malicious attacks can be further classified into two different classes.

Blind: A malicious attack is said to be blind if it tries to remove or make the watermark unrecoverable without exploiting knowledge of the particular algorithm that was used for watermarking the asset. For example, copy attack that estimates the watermark signal with aim of adding it to another asset.

Informed: A malicious attack is said to be informed if it attempts to remove or make the watermark unrecoverable by exploiting knowledge of the particular algorithm that was used for watermarking the asset. Such an attack first extracts some secret information about the algorithm from publicly available data and then based on this information nullifies the effectiveness of the watermarking system.

Examples of malicious attacks:

Printing and Rescanning

Watermarking of Watermarked Image (re-watermarking)

Collusion: A Number of authorized recipients of the image should not be able to come together (collude) and like the differently watermarked copies to generate an un-watermarked copy of the image (by averaging all the watermarked images).

Forgery: A Number of authorized recipients of the image should not be able to collude to form a copy of watermarked image with the valid embedded watermark of a person not in the group with an intention of framing a 3rd party.

IBM Attack: It should not be possible to produce a fake original that also performs as well as the original and also results in the extraction of the watermark as claimed by the holder of the fake original.

SWICO (Single Watermarked Image Counterfeit Original) Attack: SWICO constitutes of an attack whereby there is a possibility of reverse engineering the watermarking process, i.e. the possibility of building a fake original asset and a fake

watermark such that the insertion of the fake watermark within the fake original asset produces a watermarked asset, which is equal to the initial one as shown in equation 1.

$$E^{-1}(A) = \{A - w_f, w_f\} \quad \text{Eqn.2.1}$$

2.2.1.2 Non-Malicious: An attack is said to be non malicious if it results from the normal operations that watermarked data or any data for that matter has to undergo, like storage, transmission or fruition. The nature and strength of these attacks are strongly dependent on the application for which the watermarking system is devised. For example lossy-compression, geometric and temporal manipulations digital to analogue conversion, extraction of asset fragments (cropping), processing aimed at enhancing asset (e.g. noise reduction), etc.

Examples of Non-Malicious Attacks:

Lossy Compression: Many compression schemes like JPEG and MPEG can potentially degrade the data's quality through irretrievable loss of data.

Geometric Distortions: Geometric distortions are specific to images and videos and include such operations as rotation, translation, scaling and cropping.

Common Signal Processing Operations: They include the following:

D/A Conversion

A/D Conversion

Resampling

Re-quantisation

Dithering Distortion

Recompression

Linear Filtering Such as High Pass and Low Pass Filtering

Non-Linear Filtering Such as Median Filtering

Color Reduction

Addition of a Constant Offset to the Pixel Values

Addition of Gaussian and Non Gaussian Noise

Local Exchange of Pixels

2.2.2 On the Basis of Attacking Target:

Attacks can also be classified on the basis of whether they affect the watermarked data or

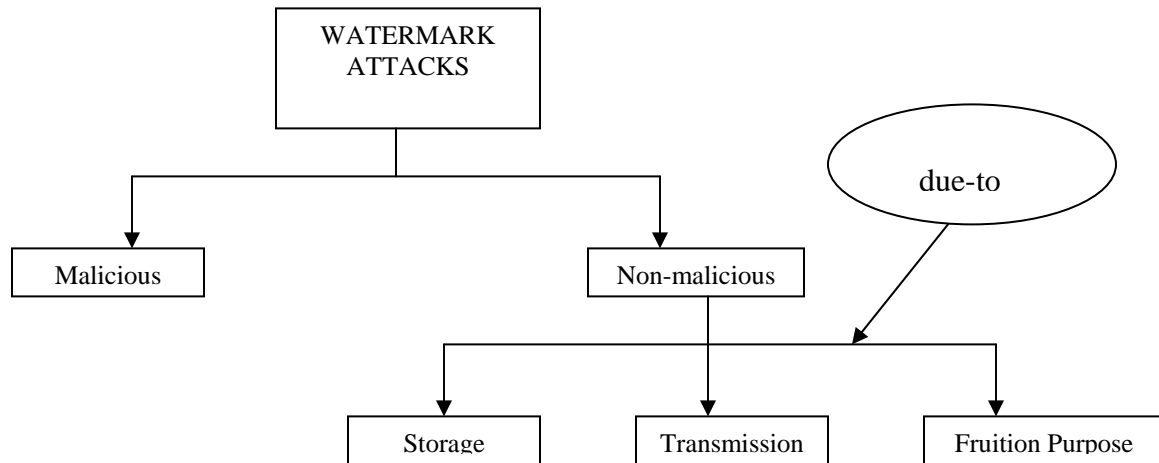


Figure 2.1 General Classification of Watermark Attacks on the Basis of Intent

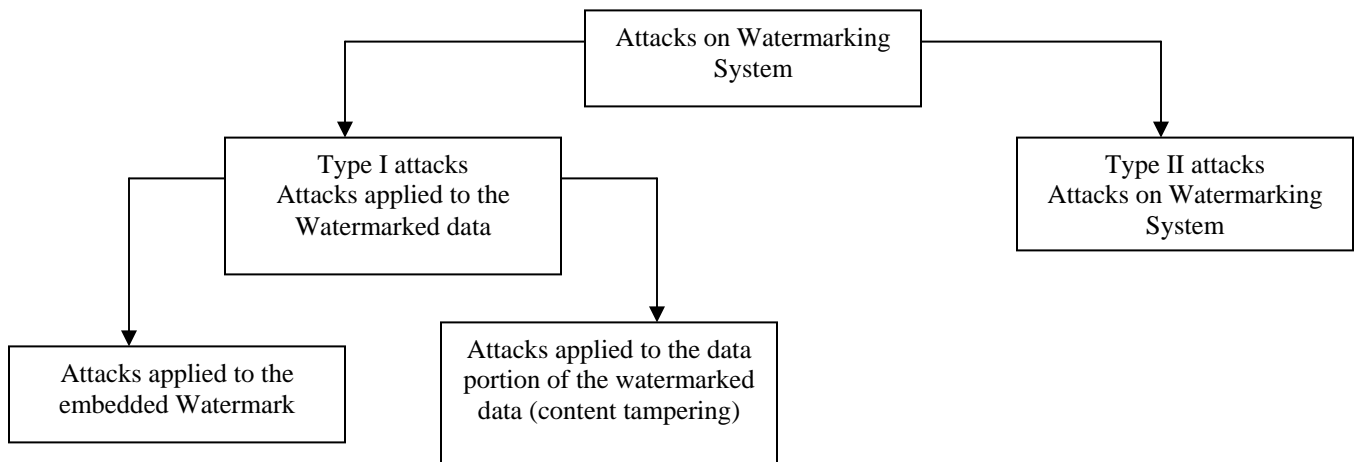


Figure 2.2.General Classification of Watermark Attacks on the Basis of Attacking Target

the watermarked system, (Fig. 2.2.)

The distinction between the two types is in the target, which each attack class focuses on. One category attack operates on the watermarked data and the other category operates on the watermarking system.

2.2.2.1 Attacks Applied to the Embedded Watermark, Type I Attacks: these usually involve some signal processing operation. This is further divided into two categories Attacks applied to the embedded watermark, which aim at making a corresponding watermark detector unable to detect the embedded watermark.

Attacks applied to the data portion of the watermarked data. These try to modify or otherwise tamper with the data in which the watermark is embedded, without destroying the watermark.

2.2.2.2 Attacks Applied to the Watermarking System Type II Attacks: These attacks may be performed without regard of the watermarked data or the original (un-watermarked) data. Therefore a signal processing operation may not be needed. This attack is usually referred to as “hacking” when it deals with software or “hardware tampering ” if it deals with hardware.

2.3 Measuring Attack Strength:

In order to compare systems based on different embedding and recovery rules, and operating in different host domains, a set of common objective, parameters must be defined [54]. False and missed detection probability and bit error rate have been used as basis for comparison.

Three parameters have been coined to measure the strength of the attack

1. **Data to Watermark Ratio (DWR):** DWR expresses the ratio between the power of the host features and that of the watermark. The rationale for using the ratio between watermarked-induced distortion and host signal power, instead of an absolute measure such as MSE, relies on the widely diffused opinion that a strong signal can accommodate a stronger watermark without compromising invisibility.

For example:

Let f be the set of features and w' the embedded watermark signal, the DWR figure is

defined as :

$$DWR = \frac{\frac{1}{n} \sum_{i=1}^n (f_i)^2}{\frac{1}{n} \sum_{i=1}^n (w'_i)^2} \quad \text{Eqn.2.2}$$

DWR depends on the asset at hand and the particular watermark that has been embedded. In order to get a global measure of algorithm obtrusiveness, the average power is considered for both the asset features and for the watermark signal, by averaging the numerator and the denominator over all possible host asset and watermarks, yielding:

$$DWR = \frac{\sum_{i=1}^n E[f_i]^2}{\sum_{i=1}^n E[(w'_i)^2]} \quad \text{Eqn.2.3}$$

If f and w' can be assumed to be stationary sequences , equation 2.2 can be simplified leading to

$$DWR = \frac{E[f^2]}{E[w'^2]} = \frac{\sigma_f^2}{\sigma_{w'}^2} \quad \text{Eqn.2.4}$$

where, the second equality follows holds under the simplified assumption that both the host features and the watermark have zero mean. DWR does not coincide with the true asset to watermark power ratio, since the average power of the host features does not necessarily coincide with the asset power. For example systems embedding the watermark in a subset of host features like a subset of DFT coefficients.

2. Watermark to Noise Ratio (WNR): WNR gives the ratio between the power of w' and that of the noise introduced by attacks.

Let the feature sequence be f'_w (assuming the feature sequence is watermarked). Let the noise $n = f'_w - f_w$ be the attack noise.

$$WNR = \frac{\sum_{i=1}^n E[(w'_i)^2]}{\sum_{i=1}^n E[n_i^2]} \quad \text{Eqn.2.5}$$

If both w' and n are stationary (zero mean) sequences that can be simplified to

$$WNR = \frac{E[(w'^2)]}{E[n^2]} = \frac{\sigma_{w'}^2}{\sigma_n^2} \quad \text{Eqn.2.6}$$

Data to noise ratio: DNR is the ratio between the original data and the noise due to the watermark.

$$DNR = \frac{\sum_{i=1}^n E[f_i^2]}{\sum_{i=1}^n E[n_i^2]} \quad \text{Eqn.2.7}$$

For stationary zero mean sequences

$$DNR = \frac{\sigma_f^2}{\sigma_n^2} \quad \text{Eqn.2.8}$$

An advantage of DNR with respect to WNR is that it reflects the real scenarios better, where the maximum allowable distortion a pirate may introduce to remove the watermark, depends on the energy of the host signal rather than the energy of the watermark.

2.4 Some Real Life Scenario

2.4.1 Removal Attacks [53]: These *aim* at the complete removal of the watermark information from the watermarked data without cracking the security of the watermarking algorithm (e.g., without the key used for watermark embedding). That is, no processing, even prohibitively complex, can recover the watermark information from the attacked data. This category includes denoising, quantization (e.g., for compression), remodulation, and collusion attacks. Not all of these methods always come close *to* their goal of complete watermark removal, but they may nevertheless damage the watermark information significantly. Sophisticated removal attacks try to optimize operations like denoising or quantization to impair the embedded watermark as much as possible while keeping the quality of the attacked document high enough. Usually, statistical models for the watermark and the original data are exploited within the optimization process. When an attacker, or a group of attackers can obtain many copies of a given data set each signed with a key or different watermark, collusion attacks come to play. In such a case, a successful attack can be achieved by averaging all copies or taking only small parts from each different copy. Recent results show that a small number of different copies (e.g., about 10) in the hands of one attacker can lead to successful watermark removal.

2.4.2 Geometric Attacks [53] :In contrast to removal attacks, *geometric attacks* do not actually remove the embedded watermark itself, but intend to distort the watermark detector synchronization with the embedded information. The detector could recover the

embedded watermark information when perfect synchronization is regained. However, the complexity of the required synchronization process might be too great to be practical. For image watermarking, the best known benchmarking tools, Unzign and Stirmark [27] integrate a variety of geometric attacks. Unzign introduces local pixel jittering and is very efficient in attacking spatial domain watermarking schemes. Stirmark introduces both global and local geometric distortions. However, most recent watermarking methods survive these attacks due to the use of special synchronization techniques. Robustness to global geometric distortions often relies on the use of either a transform-invariant domain (Fourier- Melline) or an additional template, or specially designed periodic watermarks whose auto-covariance function (ACF) allows estimation of the geometric distortions. However, as discussed below, the attacker can design dedicated attacks exploiting knowledge of the synchronization scheme. Robustness to global affined transformations is more or less a solved issue. However, resistance to the local random alterations integrated in Stirmark remains an open problem for most commercial watermarking tools. The so-called random bending attack in Stirmark exploits the fact that the human visual system (HVS) is not sensitive to local shifts and affined modifications. Therefore, pixels are locally shifted, scaled, and rotated without significant visual distortion. However, it is worth noting that some recent methods are able to resist this attack.

2.4.3 Cryptographic Attacks: Cryptographic attacks [53] aim at cracking the security methods in watermarking schemes and thus finding a way to remove the embedded watermark information or to embed misleading watermarks. One such technique is brute-force search for the embedded secret information. Another attack in this category is the so-called Oracle attack, which can be used to create a non-watermarked signal when a watermark detector device is available. Practically, application of these attacks is restricted due to their high computational complexity.

2.4.4 Protocol Attacks: Protocol attacks aim at attacking the entire concept of the watermarking application. One type of protocol attack is based on the concept of invertible watermarks [47]. The idea behind inversion is that the attacker subtracts his own watermark from the watermarked data and claims to be the owner of the watermarked data. This can create ambiguity with respect to the true ownership of the data. It has been shown that for copyright protection applications, watermarks need to be noninvertible. The requirement of non-invertibility of the watermarking technology

implies that it should not be possible to extract a watermark from a non-watermarked document. A solution to this problem might be to make watermarks signal-dependent by using one-way functions. Another protocol attack is the copy attack. In this case, the goal **is** not to destroy the watermark or impair its detection, but to estimate a watermark from watermarked data and copy it to some other data, called target data [48]. The estimated watermark is adapted to the local features of the target data to satisfy its imperceptibility. The copy attack is applicable when a valid watermark in the target data can be produced with neither the algorithmic knowledge of the watermarking technology, nor the knowledge of the watermarking key. Again, signal-dependent watermarks might be resistant to the copy attack.

2.4.5 Estimation- Based Attacks: Here attacks that take into account the knowledge of watermarking technology and exploit statistics of the original data and watermark signal [45, 48-52] have been considered. For the design of attacks against watermarking schemes, the distortion of the attacked document and the success of watermark impairment is considered. Within the scope of these attacks, concept of estimation-based attacks has been presented. This concept is based on the assumption that the original data or the watermark can be estimated, at least partially, from the watermarked data using some prior knowledge of the signals statistics. The estimation does not require any knowledge of the key used for watermark embedding. Furthermore, knowledge of the embedding rule is not required, but the attack can be more successful with it. Depending on the final purpose of the attack, the attacker can obtain an estimate of the original data or of the watermark based on some stochastic criteria such as maximum likelihood (ML), maximum a posteriori probability (MAP), or minimum mean square error (MMSE). More stress is put on different ways to exploit the obtained estimates to impair the embedded watermark. The different estimates are discussed below.

2.4.5.1 Estimate of the original data: Considering the watermark as noise in the watermarked data, the attacker can try to estimate the original unwatermarked data. This attack results in the design of an optimal denoising scheme. Recent investigations have established a strong connection between denoising and compression for filtering of additive noise from the images. This means in the case of image watermarks, the attacker can easily apply the most recent advanced coders based on wavelet decomposition to

remove the watermark. Keeping in mind the design of such coders in terms of an optimal rate-distortion trade-off, the attacker can obtain a considerable gain in resolving the compromise between distortions introduced by the attack and success in removal of the watermark. By denoising and optimized compression, the perceptual and objective quality of the attacked image can be improved significantly. Hence both denoising and optimized compression can be considered as removal attacks.

2.4.5.2 Remodulation Attacks: Remodulation attacks [53] aim at modification of the watermark using modulation opposite to that used for watermark embedding. Assuming the estimated watermark is correlated with the actual watermark, meaning a good estimate could be obtained; the estimated watermark can be subtracted from the watermarked data. Subtracting a very inaccurate estimate of the watermark might decrease the document quality without affecting the watermark too much. On the other hand, correlation-based detection can be defeated by subtracting an amplified version of the estimated watermark. For this reason a gain factor $\gamma \geq 1$ has been introduced which provides the possibility to trade off the distortion of the attacked document vs. the success of the attack. There are four basic variations of the remodulation attack. First, when $\gamma = 1$, the attack yields the MMSE/MAP estimate of the original and reduces to the denoising attack. Second, for $\gamma > 1$, the quality of the attacked document might be reduced, but correlation based detection might be effected more successfully. The attack can even drive the correlation to zero so that the detector incorrectly decides that the watermark is not present in the attacked data. Third, when using a more sophisticated distortion measure than simple MSE, weighting the remodulated watermark by a perceptual mask can obtain a better compromise between success of the attack and introduced distortion. Fourth, the attacker can not only subtract the weighted, estimated watermark, but also add outliers to obtain a non-gaussian noise distribution, which decreases the performance of correlation-based detection. Moreover, exploiting features of the human perceptual system, the attacker can efficiently embed a large amount of outliers in perceptually less significant parts of the data. For image data, this approach has been demonstrated to be successful in [49]. This attack has been referred to as perceptual remodulation (Fig. 2.4).

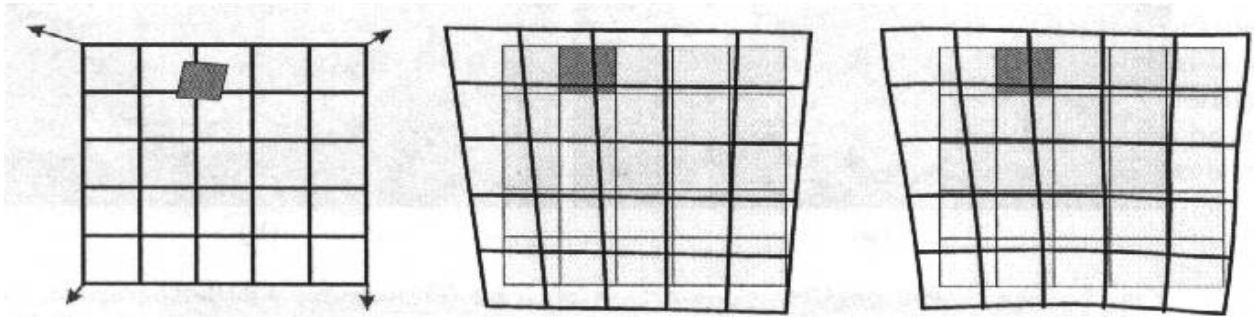


Figure 2.3 Distortion applied to still pictures by StirMark after bending and randomisation

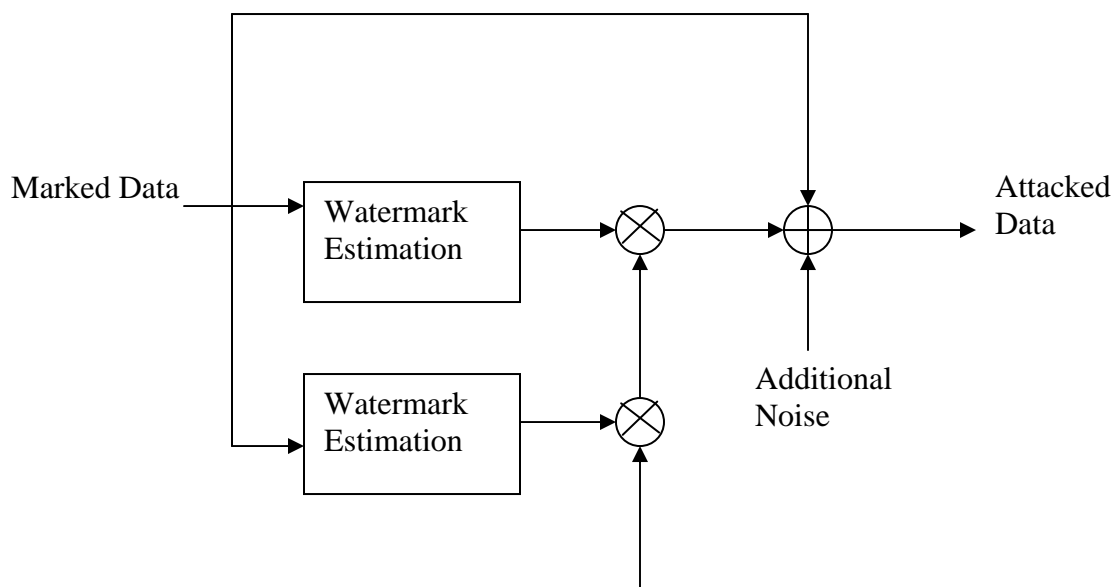


Figure 2.4 A Perceptual Remodulation Attack

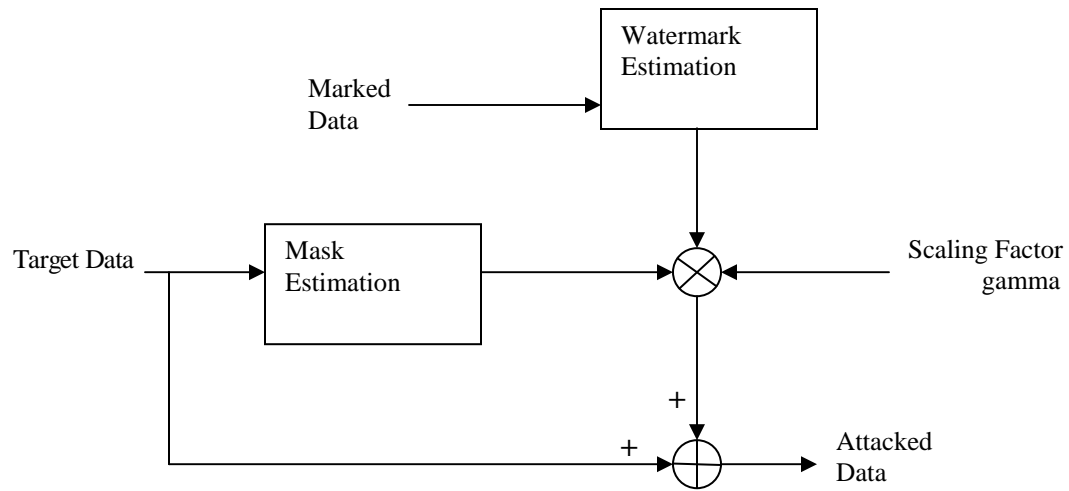


Figure 2.5 A Copy Attack

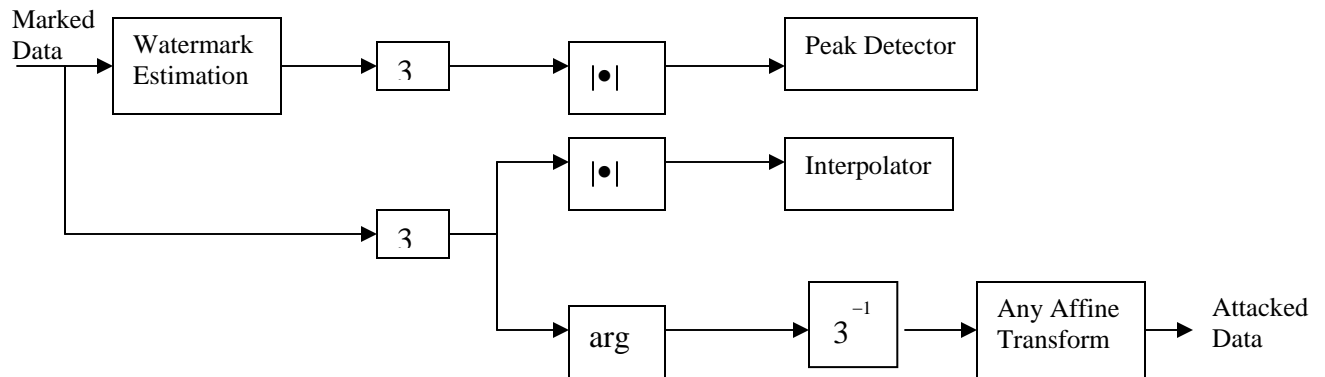


Figure 2.6 A Synchronization Removal Attack

2.4.5.3 Copy Attack: The estimated watermark can be exploited to implement a copy attack,. The copied watermark has to be adapted to the target data to keep the quality of the falsely watermarked target data high enough. There are many practical ways to adapt the watermark to the target data based on perceptual models. For images, contrast sensitivity and texture masking phenomena of the HVS can be exploited. The estimation-based copy attack is most successful when the same perceptual model is used as in the original watermarking algorithm. The copy attack in its described version is mainly applicable to additive watermarking schemes. In the case of quantization-based watermarking schemes, even a perfectly estimated watermark signal w cannot be copied since it is highly unlikely that the copied signal w is a valid watermark in the target signal (Fig. 2.5).

2.4.5.4 Synchronization Removal: Watermark estimation can also be very efficiently applied to attack synchronization mechanisms. The basic idea of synchronization removal is to detect synchronization patterns, remove them, and then apply desynchronization techniques, such as global affine transformation in the case of image watermarking. Here, a synchronization method for image watermarking based on a template in the magnitude image spectrum or on the ACF of periodic watermarks have been considered. In both cases, peaks are generated in the Fourier domain [45,49]. It is obvious that such peaks can easily be detected. Once the peaks have been detected, the next step of the attack is to

interpolate the spectrum of the watermarked image or previously attacked image in the locations of spatial frequencies determined by a local peak detector. The generalized block diagram of this attack is shown in (Fig.2.6) where \mathcal{F} and \mathcal{F}^{-1} denote direct and inverse Fourier transforms, respectively, $|\bullet|$ is the magnitude, and \arg is the phase.

2.4.5.5 Benchmark Including Estimation Based Attack

The Stirmark benchmark is an excellent tool for measuring the robustness of watermarking algorithms; it is heavily weighted toward geometric transformations, which does not take into account prior information about the watermark. Therefore, another benchmark was proposed [45], which contains the following six categories of attacks:

1. Denoising: Wiener filtering, soft shrinkage, and hard threshold.
2. Denoising followed by perceptual remodulation.



Figure 2.7 a) Original Lena Image



Figure 2.7 b) Watermarked work
Detector output 94.6641



Figure 2.7 c) After StirMark Attack
Detector output 1.7644

3. Denoising followed by Stirmark random bending.
4. Copy attack: The watermark is estimated using Wiener filtering and copied onto another image. If the watermark is successfully detected in the new image, the algorithm has failed.
5. Template removal followed by small rotation.
6. Compression based on wavelet decomposition of the image, which is superior to JPEG compression and is integrated to reflect the future appearance of the JPEG2000 standard. *Also* included in this benchmark is a new proposal for evaluating image quality. Rather than using PSNR, two approaches based on weighted PSNR and Watson's metric are proposed. The Watson metric measures local image characteristics and reflects luminance, contrast, and texture masking.

2.5 Counter Measure Against Estimation Based Attacks:

General Precautions:

- Number of pixels that one bit of watermark information is distributed over should not be too low
- Avoidance of cryptological weaknesses: used keys should be secure (impossible to be determined)
- Use of collusion-secure watermarks
- Use of non-invertible watermarks: signal-adaptive watermarks (code depends on host data)
- Adaptation of the watermark power spectrum: watermark should have strong frequencies where host data also has them.
- Use of counterattacks drawback: possible attacks have to be foreseen

General Counterattacks:

- Preventing Unauthorized Embedding (for MM Data Authentication Purpose)
Cryptographic tools or digital signature should be used to prevent forging. To counter copy attack the watermark should be made host data dependent.
- Preventing Unauthorized Detection
Before embedding the watermark it should be encrypted so that decoding should include decrypting.

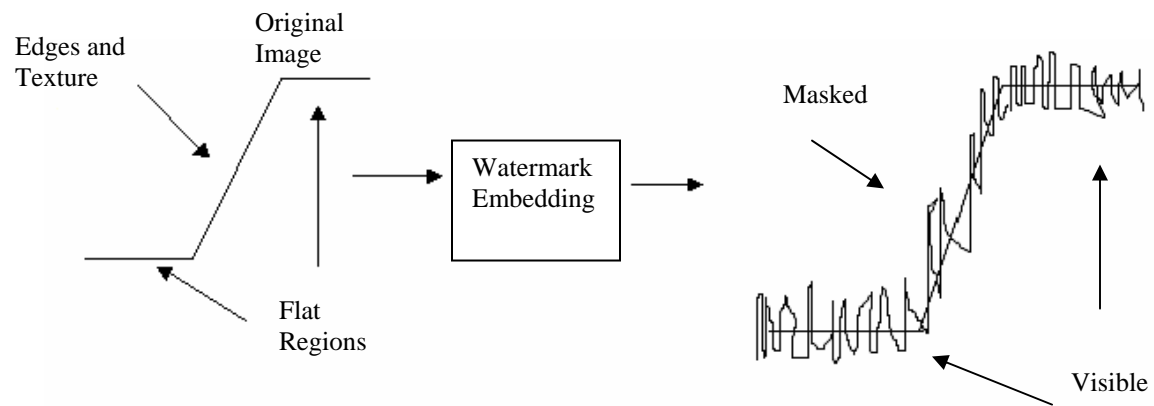


Figure 2.8 a) The Data Hider Strategy Exploiting the Texture Masking Function of HVS;

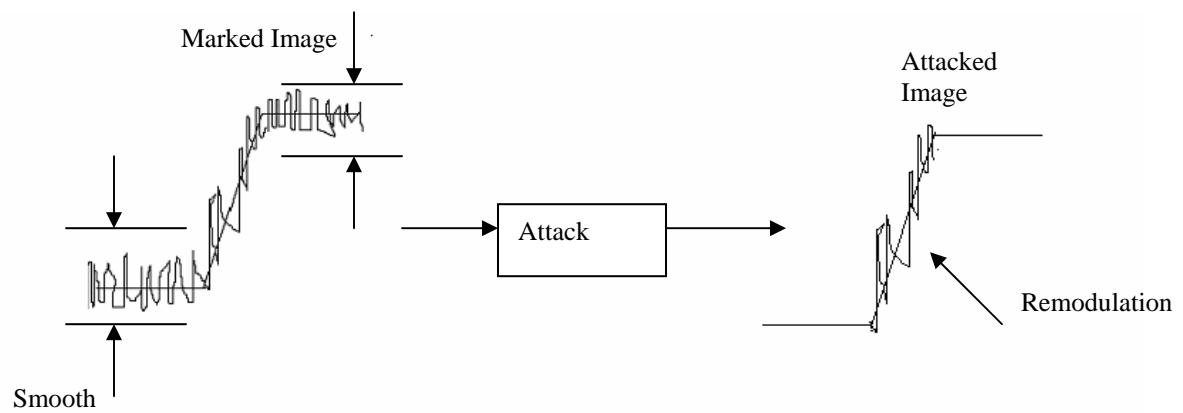


Figure 2.8 b) The Attacker Strategy Using Denoising and Perceptual Remodulation

- Preventing Unauthorized Removal
Depends on specific attack

To resist estimation-based attacks, the embedder aims at making the watermark difficult to estimate. This approach has been investigated for two different scenarios in [54].

2.5.1 Power-Spectrum Condition - An idealized theoretical approach [50] for analyzing estimation-based attacks treats the original signal and watermark as independent, zero-mean, stationary, colored Gaussian random processes. The watermarked data is the sum of these two processes. The original signal is given so its power spectrum is fixed, but the watermark power spectrum can be varied. The watermark power spectrum should be shaped to resist an estimation-based attack. For this scenario, the optimal estimate is obtained by a Wiener filter. The MSE E between the original watermark and the estimated watermark provides a convenient way to measure how well a watermark resists estimation. It has been shown that E is maximized if and only if the watermark power spectrum is directly proportional to the power spectrum of the original signal. This requirement is called the power-spectrum condition (PSC). A watermark whose power spectrum satisfies the PSC is the most resistant against estimation. If distortion is measured by the mean squared difference between the attacked data and the unwatermarked original data, the PSC has another important consequence: For any output of the matched filter, a watermark that fulfills the PSC causes the above attack to incur the greatest distortion. To drive the correlation to zero, the attack must make the distortion as large as the power of the original data, so the attacked data is unlikely to be useful.

2.5.2 Noise Visibility Function - The PSC is attractive because it can be proven rigorously and has a convenient mathematical form. For image watermarking, image denoising provides a natural way to develop estimation-based attacks [53] optimized for the statistics of images, although optimality might be difficult to prove. The watermarked image is treated as a noisy version of the original image, and the watermark represents noise that should be eliminated. Thus, the estimated watermark is the same as the estimated noise. In [54] Mauro Barni et al have applied different statistical models for the original images, namely a non-stationary Gaussian process or a stationary, generalized Gaussian process. The noise/watermark can be treated as one of these processes. Assuming that it is still a stationary Gaussian process. In the first case, the

denoising method uses an adaptive Wiener filter, while in the second it reduces to the popular denoising methods of hard-threshold and soft-shrinkage as particular cases. Both denoising methods produce a *Texture masking function* (TMF), which is derived from the image statistics and is therefore image dependent. The TMF takes on values in [0,1]. To embed a watermark that resists such estimation, the watermark embedding should use the inverted function known as a noise visibility); *function* (NVF), defined by $NVF = 1 - TMF$. NVF values near zero indicate flat regions, where the watermark should be attenuated; while NVF values near one indicate texture or edge regions, where the watermark should be amplified. In this way, the watermark is embedded to resist estimation-based attacks derived from image denoising. A qualitative comparison sheds light on the structure of watermarks produced in this manner. PSC can give only a coarse result since the underlying statistical model does not fit very closely to images. Nevertheless, the results obtained using the PSC agree with those of the NVF. The two approaches complement each other well. These results support the heuristic argument of *Cos et al.* [42] that the watermark should be placed in the “perceptually significant frequency components.”

2.6 Conclusion

The various attacking algorithms in the image-processing domain were discussed. Parameters that measure the obtrusiveness of a watermark were presented. Finally some real life scenarios along with some counter measures were presented. However algorithms specifically for the purpose of attacking a watermark have not been modeled in this work. In chapter 5 non-malicious attacks like low pass filtering and compression have been modeled and applied to the proposed algorithm to test its robustness.

CHAPTER 3

LITERATURE REVIEW

OBJECTIVE: Study of already existing work in literature

CHAPTER ORGANISATION:

- 3.1 Introduction
- 3.2 Algorithms in Spatial Domain
- 3.3 Advantages of Frequency Based Methods
- 3.4 Algorithms in Frequency Domain
- 3.5 Advantages of Compressed Domain
- 3.6 Algorithms in Compressed Domain
- 3.7 Conclusion

3.1 Introduction

In 1994 Van Schyndel et al [102] changed the LSB of an image to embed an m_sequence watermark. Since then, more and more researchers have studied digital watermarking problem.

In general, watermark can be embedded in spatial domain or transform domain or compressed domain of an image. Spatial domain techniques directly modulate the pixels. In the spatial domain approach, such as van Schyndel et al [102], Pitas et al, [41,103-105,111], J.Zhao and E. Koch [41], K. Hirosugu [55], R. G. Wolfgang and E. J. Delp, [84,107], M. Schneider and S. F. Chang [106], Mintzer et al [108,109], Braudaway [78], M.Kutter et. al [110], Hernandez et al [112], Wu and Tsai [113], W. Bendor, et. al [114], J.F.Delaiglee et. al [115], Bangaleea and Rughooputh [116], Dipti et al [117] the pixel value of an image is modified to embed watermark information.

Transform domain techniques modify the DCT, DWT or DFT or any other transformed coefficients. Transform domain techniques usually achieve better performance since the perceptual characteristics of images can be better utilized and the spread spectrum principles used in secure communications can be easily incorporated [42]. Typically transform domain systems perform the watermarking process independent of compression, although these two processes share some common features. Wolfgang et al [118] have investigated whether matching the watermarking domain to compression increases the robustness.

Compressed domain techniques integrate the compression framework with watermarking by directly labeling the compressed (quantized) symbol streams. There is little loss in generality by assuming a compressed domain framework since compression is nearly ubiquitous for multimedia.

3.2. Algorithms in Spatial Domain

P.G.Van Schyndel et al. [102] discuss two methods. The first is based on bit-plane manipulation of the LSB, which offers easy and rapid decoding. The second method utilizes linear addition of the watermark to the image data and is more difficult to decode,

offering inherent security. The first method involves the embedding of the m-sequence on the LSB of the image data. The second method uses LSB addition for embedding the watermark. The decoding process makes use of the unique and optimal auto-correlation function of m-sequences. The process requires the examination of the complete bit pattern and its current implementation, must therefore be performed off-line which is principal disadvantage. However, it is intrinsically more secure, since a potential code breaker has to perform the same operation, without any a priori knowledge. The main problem found with adding the watermark is in retaining the dynamic range of the original image and the auto-correlation output. The watermark is robust to averaging, and potentially compatible with JPEG compression.

J.Zhao and K.Koch [41] developed the product Syscop, which is compatible with JPEG compression quality factor 50%. An image is partitioned into 8x8 blocks and eight coefficients in the blocks are used for marking. The blocks are pseudo randomly selected to minimize detection. The method is still weak against physical damages (e.g. cut a pixel live, grab an area etc.). M.Scheinder and S.F.Chang [106] present a variation on J.Zhao and K.Koch [41] method for image authentication. The general procedure for generating a constant based signature is given as follows:

- The content of interest C_o is extracted from the original image I_o using extraction function F_o .
- The content is hashed using a hash function F_h to reduce the amount of data.
- The hash H_o is then encrypted using the private key K_{pr} of the signing entity to produce the final signature S . Mathematically it can be written as follows:

$$\left. \begin{array}{l} C_o = F_c(I_o) \\ H_o = F_h(C_o) \\ S = H_o + K_{pr} \end{array} \right\} \quad \text{Eqn.3.1}$$

To verify the authenticity of an questionable image I_t , the following steps are followed:

- The signature is decrypted using the public key K_{pu}

- Then it is compared with the hashed content extracted from the questionable image.
- If the distance between the feature vectors is less than a threshold value t , then the questionable image is declared un-manipulated. Mathematically we can write the step as below:

$$\left. \begin{array}{l} C_t = F_c(I_t) \\ H_t = F_h(I_t) \\ \|H_o - H\| < t \end{array} \right\} \quad \text{Eqn.3.2}$$

The algorithm is compatible with certain types of image modification (e.g. lossy compression) R.B.Wolfgang and E.J.Delp [84] present a watermark, which is a two dimensional extension of [102]. Their first watermark is robust to the mean and median filtering and the second watermark is robust to JPEG compression. The first watermark uses a much longer m-sequences than its counterpart [102], which is, arranged row by row into two-dimensional blocks. Then a zero is appended to the entire m-sequences, instead of using an extended m-sequence. One advantage of a two-dimensional watermark is the ability to more effectively locate where an image has been changed. They define the spatial cross correlation function of images X and Y as:

$$R_{xy}(\alpha, \beta) = \sum_i \sum_j X(i, j)Y(i - \alpha, j - \beta) \quad \text{Eqn.3.3}$$

Let X be the original image block, W be the watermarked block, Y be the water marked image block and Z be the watermarked image block that might be forged. The test statistics for the block, is defined as:

$$\delta = R_{yw}(0,0) - R_{zw}(0,0) \quad \text{Eqn.3.4}$$

If the watermarked image is unchanged, then $\delta = 0$. When δ is larger than a defined tolerance the block fails the watermark test. A larger threshold provides more robustness but increases the probability of missing a forgery the authors revised this watermarking technique to improve security and localization. Localization is the ability to identify where in the image any changes have occurred. The block size is 8x8 pixels and each block is formed as follows:

- A large span m-sequence ($n = 96$) is generated with the first 128 bits skipped.

- The next 64 bits are inserted in the first block of the watermark column by column. The next m bits are skipped.
- The procedure repeats for the remaining blocks. To make it JPEG compatible new statistics was defined:

$$\delta = R_{y'w}(0,0) - R_{z'w}(0,0) \quad \text{Eqn.3.5}$$

where Y_j is the watermarked image after JPEG compression and decompression and Z_j be possibly forged water marked image after JPEG processing. R.B.Wolfgang and E.J.Delp [107] elaborated watermarking techniques introduced in [84]. They describe how their techniques withstand random errors. Here they find the mean of d for all blocks as follows:

$$E[|\delta|] = \frac{1}{N} \sum_i^N \delta_k \quad \text{Eqn.3.6}$$

where δ_k is the value of d for the k^{th} block and N is the number of 8x8 blocks in the mage. The test statistics $E[|\delta|]$ is not robust to small attenuation as it is very small. This is a potential problem. They give the range of $E[|\delta|]$ for an image to be fully authentic but forged, possibly authentic and completely in authentic (or water marked by a different owner).

K.Hirotsugu [55] presents a digital signature system to assert copy right of image data. The outline of the signature method is shown as follows:

- The original image is divided into 8x8 blocks.
- The region graph is generated.
- A random permutation is generated as secret information of the owner of the original image.
- The concealed graph is generated.
- The graph data is concealed in the original image.

The verification process is given as follows:

- The region graph is generated.
- The concealed graph is decoded from the concealed information.
- The isomorphism between the two graphs is shown using ZKIP (zero knowledge interactive proof). This technique has a very good advantage that the secret

information is not disclosed even during the authentication process. The author has not mentioned about robustness of the technique for various attacks.

I. Pitas and N.Nikolaids and Kaskali [41,103,104,111] use an approach that allows slightly more information to be embedded. A binary signature that consists of equal no of zeros and ones is embedded in an image by assigning pixels into one of the two sets. The intensity levels of pixels in one of the sets are altered. The intensity levels are not changed in the other set. Signature detection is done by comparing mean intensity value of the marked pixels against that of the not marked pixels. Statistical hypothesis testing is used for this purpose. The signature can be designed in such a way that it is resistant to JPEG compression and low pass filtering. According to the authors, the degree of certainty can be as low as 84% and as high as 92%, which would likely not stand up as evidence in a court of law for copyright protection. G.Voyatris and I.Pitas [105] use toral automorphism to chaotically mix binary logos or signatures, which are added to a secret region in the image. The embedding algorithm is robust to noise filtering and compression. The reconstructed watermark is recognized visibly if the watermarked is affected by JPEG compression up to 6:1. By using detection methods we can get a reliable answer about the existence or not of a watermark even if the watermarked image has been affected quite strangely by filtering and JPEG compression, greater than 10:1. mpression).

M.M.Yeung and F.Mintzer [109] have proposed an invisible image watermarking technique for image verification, where one is interested in knowing whether the content of the image has been altered since some earlier time, perhaps because of the act of a malicious party. That means that this is an invisible fragile watermarking algorithm. In the watermarking process, a binary map $B(i,j)$ of a watermarked image $W(i,j)$, is embedded into the source image $I(i,j)$, to produce a stamped image $I'(i,j)$. In the image verification processing, a watermark image, $B(i,j)$, is extracted from a stamped image $I'(i,j)$. The watermarking process does not introduce visual artifacts and retain quality of the images. A verification key is produced together with a stamped image. The embedded watermark image can be extracted from the stamped image using this verification key. Alternations to an image introduce artifacts on the extracted image, which can be visually and automatically identified. The technique offers fast image verification to detect and

localize unauthorized image alterations. The technique provides a means of ensuring data integrity, adds to security of digital content and allows the recipients of an image to verify the image as well as to display the ownership information on the image.

Kutter et al. [110] has presented a spatial domain method which does not need the original for the purpose of extraction of the watermark. Signal bits are multiplied and embedded by modifying the pixel values in the blue channel, the color that the eye is least sensitive to. Further the modifications are additive, subtractive depending on the value of bit, the proportional to the luminance. For extraction the prediction of the original pixel value is used in place of the original. A cross-shaped neighborhood is chosen to predict the same. The sign of the difference of the predicted value and the pixel value of the watermarked copy is used in decision making. Random locations are selected for embedding. Two bits are always maintained as 0 and 1 which define a geometric reference used for countering geometrical attacks. An adaptive threshold is also used to improve decision making. The scheme is robust to blurring, JPEG compression (75% quality factor), and geometrical operation.

J.R.Hernandez et al. [112] model and analyze a watermarking scheme for copyright protection. In this scheme a signal following a key-dependent 2-D multiple modulation is added to the image for ownership enforcement purpose. They derive bound and approximations to the receiver operating characteristic. The results can be used to determine the threshold associated to a required probability, false probability of detection. They model the data-hiding process as communication channel.

Bender et al. [114] proposed a patchwork in which a watermark is embedded into the image by modifying the statistical property of the image. The difference between any pair of randomly chosen pixels is Gossip distributed with a mean zero. This mean can be shifted by selecting pair of points and incrementing the intensity of one of the points and decrementing the intensity of the other. The resulting watermark is predominantly high frequency. However the authors recognize the importance of placing the watermark in perceptually significant regions and consequently modify the approach so that pixel patches rather than individual pixels are modified, thereby shaping the watermark more to significant regions of the human visual system. Patchwork is robust to cropping but suffers from the disadvantage of being highly sensitive to affine transformation, JPEG

compression & very low bit rate i.e. one bit per image. The inventors provide data that the recovery rate is 85% after JPEG compression, with quality parameter 75%, which is not likely stand up as credible evidence beyond a reasonable doubt in a course of law. Bender et al. [114] have also proposed texture block coding in which a block from a region with a random texture is copied and placed in a region with similar texture. Detection of hidden blocks is easy and can be done as follows:

- The image is auto correlated with itself. This will produce peaks at every point in the autocorrelation where identical regions of the image overlap. If large enough areas of an image are copied, this will produce an additional autocorrelation peak at the correct alignment for coding.
- The image is shifted as indicated by the peaks in previous step. Now, the image is subtracted from the shifted copy, padding the edges with zeros as needed.
- The result is squared and threshold to recover only those values quite close to zero. The copied region will be visible at these values. The method requires a human operator to choose the source and destination regions and to evaluate visual impact of the modifications on the image. The technique will not work on the images that lack moderately large areas of continuous texture from which to draw.

J.F.Delaiglee et al. [115] present an additive watermarking technique for grey-scale images. It contains in secretly embedding a binary code into the image without degrading its quality. Those bits are encoded through the phase of maximal length sequences (MLS). The core of the embedding process is under laid by a masking criterion that guaranties the invisibility of the watermark. It combined with an edge and texture discrimination to determine the embedding level of the MLS, where bits are actually spread over 32x8 pixel blocks. The watermarking method is resistant to white noise, JPEG compression, low pass filtering and forgery.

R. Bangaleea and H.C.S. Rughooputh [116] have proposed an algorithm, where a small number of bits are embedded onto an image in the spatial domain using a method similar to the direct sequence spread spectrum. A N-bit long information $B = \{b_1, b_2, \dots, b_n\}$ is embedded in an image I in the brightness plane of the image. The message bits are

modulated with a PN sequence by Spread Spectrum modulation so that the watermark is tamper proof and has anti-jam properties in the transmission channel. Perceptual analysis is done with a masking function to determine regions of the image, which are perceptually significant. To find good compromise between robustness and imperceptibility, the mask was scaled. The watermarking method produced good results for contrast modification, linear filtering, blurring, gamma correction, brightness modification, rescaling and cropping.

In [117] Dipti Prasad Mukherjee et al have presented an invisible spatial domain watermark insertion algorithm for which we show that the watermark can be recovered, even if the attacker tries to manipulate the watermark with the knowledge of the watermarking process. The process incorporates buyer specific watermarks within a single multimedia object, and the same multimedia object has different watermarks that differ from owner to owner. Therefore recovery of this watermark not only authenticates the particular owner of the multimedia object but also could be used to identify the buyer involved in the forging process. This is achieved after spatially dividing the multimedia signal randomly into a set of disjoint subsets (referred to as the *image key*) and then manipulating the intensity of these subsets differently depending on a buyer specific key. These buyer specific keys are generated using a secret permutation of error correcting codes so that exact keys are not known even with the knowledge of the error correcting scheme. During recovery process a manipulated buyer key (due to attack) is extracted from the knowledge of the image key. The recovered buyer key is matched with the exact buyer key in the database utilizing the principles of error correction. The survival of the watermark is demonstrated for a wide range of transformations and forging attempts on multimedia objects both in spatial and frequency domains. We have shown that quantitatively our watermarking survives rewatermarking attack using the knowledge of the watermarking process more efficiently compared to a spread spectrum based technique. The efficacy of the process increases in scenarios in which there exist fewer numbers of buyer keys for a specific multimedia object. We have also shown that a minor variation of the watermark insertion process can survive a “Stirmark” attack. By making the image key and the intensity manipulation process specific for a buyer and

with proper selection of error correcting codes, certain categories of collusion attacks can also be precluded.

3.3 Advantages of Frequency Based Methods

Lossy compression is an operation that usually eliminates perceptually non-salient components of an imager. If one wishes to preserve watermark in the face of such an operation, the watermark must be placed in the perceptually significant region of the data. Most processing of this sort takes place in the frequency domain. In fact, data loss usually occurs among the high frequency components. Hence the watermark must be placed in the significant frequency components (low frequency component) of the image spectrum. In contrast to the spatial-domain-based watermarking, frequency-domain-based techniques can embed more bits of watermark and are more robust to attack; thus, they are more attractive than the spatial-domain-based methods. Cox *et al.* [55] used the spread spectrum communication for multimedia watermarking. They embedded a set of independent and identical distributed sequences drawn from a Gaussian distribution into the perceptually most significant frequency components of an image. Hsu and Wu [56] embedded the watermarks with visually recognizable patterns in the images. The embedding positions were selectively modifying the middle frequency of DCT of the images. The embedding and extracting methods of the DCT-based approach have been described [56,64]. On the other hand, several methods [57–67] used the discrete wavelet transform (DWT) to hide data to the frequency domain to provide extra robustness against attacks.

3.3 Algorithms in Frequency Domain

J.J.K.O’Runaidh *et al.* [68] present a perceptual watermarking method operating in the transform domain. They argue that water marking needs to be adaptive in order to be robust and place the watermark in the perceptually most significant components of the image. A watermark is non-intrusive if it resembles the image that it is designed to protect. That means less information should be hidden on flat featureless regions of the image & more information in the parts of the image that contain more texture or around edges, provided edge integrity is maintained. Transform domain methods is preferred as it is felt that it is possible to mark according to perceptual significance of different transform components and the watermark gets regularly distributed over the entire image

sub-block, making it more difficult for attackers in possession of independent copies of the image to decode and read the mark. The watermark survived 20:1 JPEG compression on the standard 256x256 Lena image. J.J.K.O’Runaidh et al. [69] prepare a discrete Fourier transform phase based method of conveying watermark information. They used the fact that phase is more important than magnitude of the DFT values. The watermark survived 15:1 JPEG image compression. M.D.Swason et al. [70] introduce a watermarking scheme for image which exploits the human vision system (HVS) to guarantee that embedded watermark is imperceptible. The insertion of watermark involves following steps:

- The image is segmented into blocks.
- DCT of each block is found.
- A frequency mark of each block is computed.
- The resulting perceptual mark is scaled and multiplied by the DCT of a minimal length pseudo-noise sequence (author id).
- The watermark is then added too the corresponding DCT block.
- The watermark image is obtained by assembling the inverse DCT of each block. They use a model for frequency masking. Spatial masking is used to verify that the watermark designed with the frequency-masking model is invisible for local spatial regions. Detection of the watermark is accomplished via hypothesis testing. Experimental results show that the watermark is robust to several distortions including white and colored noise, JPEG coding at different qualities and cropping.

I.J. Cox et al. [71,81] propose to insert a watermark into the spectral components of the data using techniques analogous to spread spectrum communication. The argument is that the watermark must be perceptually significant components of a signal if it is to be robust to common signal distortions and malicious attack. However, the modification of these components can lead to perceptually degradation of the signal. The watermark insertion consists of following steps:

- DCT of the image is computed
- The perceptually significant regions of the image are found out.
- The watermark $X = x_1, x_2, \dots, x_n$ is constructed where each x_i is chosen according to $N(0,1)$, where $N(0,1)$ denotes a normal distribution with mean 0 and variance 1.

- The watermark is inserted in the DCT domain of the image by setting the frequency components v_i in the original image to V_i' using Eqn.3.7.

$$v_i' = v_i(1 + ax_i) \quad \text{Eqn.3.7}$$

where a is a scalar factor. The author chose $a = 0.1$. A Gaussian type of watermark is used because it is more robust to tampering than uniform embedding. Extraction of watermark consists of following steps:

- DCT of watermark image is computed.
- DCT of the original image is computed.
- The difference of the two is the watermark X^* . The extracted watermark X^* is compared with the original watermark X using similarity function given below

$$\text{sim}(X, X^*) = \frac{X \cdot X^*}{\sqrt{X^* \cdot X^*}} \quad \text{Eqn.3.8}$$

The watermark is robust to common signal and geometric distortion such as A/D and D/A conversion, resampling, quantization, compression, rotation, translation, cropping and scaling. The watermark is universal in the sense that it can be applied to all three media. Retrieval of the watermark unambiguously identifies the owner and the watermark can be constructed to make counterfeiting almost impossible.

C.T.Hsu and J.L.Wu [56] propose an image authentication technique by embedding each image with a signature so as to discourage unauthorized copying. A DCT based algorithm is used to implement the middle band embedding. The proposed technique could actually survive several kinds of image processing and the JPEG lossy compression.

A.G.Bors and I. Pitas [72] propose an algorithm for image copyright protection. The algorithm proposed selects certain blocks in the image based on a Gaussian network classifier. The pixel values of the selected blocks are modified such that their DCT coefficients fulfill a constraint imposed by the watermark code. Two different constraints are considered. The first approach consists of embedding a linear constraint among selected DCT coefficients and second one defines circular detection regions in the DCT domain. The watermark detection is based on the probability detection theory. The proposed algorithm is resistant to JPEG compression.

C.Podilchuk and W.Zeng [73] propose a watermarking technique for digital images that is based on utilizing visual models, which have been developed in the context of image

compression. The visual models give a direct way to determine the maximum amount of watermark signal that each portion of an image can tolerate without affecting the visual quality of the image. The scheme is best suited for destination faced. The watermark encoding scheme consists of a frequency decomposition based on an 8x8 framework followed by JND calculation and watermark insertion. Watermark detection is based on classical detection theory. The original image is subtracted from the received image and the correlation between the signal difference and specific watermark sequence is determined. The maximum correlation value is compared to a threshold to determine whether the recovered contains the watermark in question. The watermarking scheme is extremely robust to JPEG compression, cropping, scaling, additive noise, gamma correction and the combination of printing/Xeroxing/rescanning.

A. Piva et al. [74,80] propose a method in which watermark casting is performed by exploiting the masking characteristics of HVS and the embedded sequence is extracted without reporting to the original image. The watermark insertion process involves the following steps:

- N X N DCT of an N X N image is computed.
- DCT coefficients are recorded into a zigzag scan.
- The first $L + M$ coefficients are selected to generate a vector

$T = \{t1, t2, \dots, tL, tL+1, \dots, tL+m\}$. In order to obtain a trade off between perceptual invisibility and robustness to image procuring techniques, the lowest L coefficients are skipped and a watermark $X = \{x1, x2, \dots, xm\}$ is embedded in the last m members to obtain a new vector $T' = \{t1', t2', \dots, tL', tL + M'\}$ according to the following rule:

$$L + i' = tL + i + a \cdot |tL + i| \cdot xi \quad \text{Eqn.3.9}$$

where $i = 1, 2, \dots, M$. The vector T' is then reinserted in the zigzag scan and inverse DCT algorithm is performed obtaining the watermarked image I' . In order to enhance the robustness of the watermark characteristics of the HVS is exploited to adapt the watermark to the image being signed. The original image I and the watermarked image I' are added pixel by pixel according to the local weighting factor bij , obtaining a new watermarked image I'' , i.e.

$$y''_{ij} = y_{ij}(1 - b_{ij}) + b_{ij}y'_{ij} = y_{ij} + b_{ij}(y'_{ij} - y_{ij}) \quad \text{Eqn.3.10}$$

The b_{ij} takes into account the characteristics of the HVS. The watermark detection process consists of following process:

- Given a possibly corrupted image I^* , the $N \times N$ DCT is applied to I^* .
- The DCT coefficients are recorded into a zigzag scan.
- The coefficients from the $(L+1)^{\text{th}}$ to the $(L+M)^{\text{th}}$ are selected to generate a vector

$$T^* = \{t^*_{L+1}, t^*_{L+2}, \dots, t^*_{L+M}\}$$

- The correlation between the marked and the possibly corrupted coefficients T^* and the mark itself is taken as a measure of mark presence. The watermark is robust to JPEG compression, lowpass and medium filtering, histogram equalization and stretching, Gaussian noise addition, nesting cooping and multiple watermarking.

J.J.K.O'Ruanaidh and T.Pun [75,84] propose that Fourier transform based invariants can be used for digital image watermarking. The watermark takes the form of a two-dimensional spread spectrum signal in the RST transformation invariant domain. The watermark survives lossy image compression using JPEG at normal setting (75% quality factor). The watermark is also reasonably resistant to cropping and could be recovered from a segment approximately 50% of the size of the original image.

J.R.Smith and B.O.Comisky [76] use the concepts of communication theory to characterize information-hiding schemes. They introduce a framework for quantifying the tradeoffs among the conflicting figures of merit useful for characterizing information hiding schemes:

- capacity (the number of bits that may be hidden and then recovered),
- robustness to accidental removal and
- imperceptibility.

They also introduce a technique called “pre-distortion” for increasing resistance to JPEG compression. They pointed out that frequency hopping spread spectrum is superior perceptually and in terms of robustness to accidental removal, to direct sequence spread spectrum.

D.J.Fleet and D.J.Heeger [77] describe method for embedding information in color images. A model of human color vision is used to ensure that the embedded signal is invisible. Sinusoidal signals are embedded so that they can be detected without use of

original image. The embedded information is robust enough to be reliably extracted after being printed and scanned on common place equipment.

G.W.Braudway [78] presents a method for marking high quality digital image with a robust and invisible watermark. The watermark is imparted into an image as a random but reproducible, small modulation of its pixel brightness and becomes a permanent of the marked image. Watermark detection can be done without using the original image. The detection method presented exploits the not all understood but superb ability of the human visual system to recognize a correlated pattern in a scattered diagram called a “visualizer-coincidence image”. The watermarking scheme is robust to JPEG compression, printing and rescanning of the image.

C.I.Podilchuk and W.Zeng [79] have proposed two watermarking techniques for digital images that are based on utilizing visual models, which have been developed in the context of image compression. One technique is based on frequency decomposition consists of block based DCT and the second technique is based on frequency decomposition consists of a wavelet decomposition. First technique is same as already described in [73]. The second technique based on a wavelet decomposition uses almost similar insertion and detection procedure, but has got lots of advantages over its DCT counterpart as discussed here. Due to the hierarchical decomposition, this approach has an advantage of constructing watermark components that have varying spatial support providing the benefits of both spatially local and spatially global watermark. The watermark components with local spatial support are suited for local visual masking effects is and robust to signal processing such as cropping. The watermark components with global spatial support are robust to operations such as low pass filtering. DCT faced framework produces watermarks wit only local spatial support and the spread spectrum approach produces watermarks with global spatial support.

Tao and Dickinson [82] propose an additive watermarking techniques, which assigns each spatial region a noise sensitive level and embeds the watermark-using block DCT according to this sensitivity level. NAC coefficients having the smallest quantization step sizes in the JPEG quantization tables are selected and modulated. The authors have arrived at a formula based on the noise sensitivity of the block for the extent of perturbation possible to the ac coefficients. The blocks are classified into 6 perceptual

classes basing on gradient and variances in the particular blocks. The watermark generated is a function of image itself.

In [85] Juan et al have proposed a spread-spectrum-like discrete cosine transform domain (DCT domain) watermarking technique for copyright protection of still digital images. The DCT is applied in blocks of 8×8 pixels as in the JPEG algorithm. The watermark can encode information to track illegal misuses. For flexibility purposes, the original image is not necessary during the ownership verification process, so it must be modeled by noise. Two tests are involved in the ownership verification stage: watermark decoding, in which the message carried by the watermark is extracted, and watermark detection, which decides whether a given image contains a watermark generated with a certain key. Generalized Gaussian distributions is applied to statistically model the DCT coefficients of the original image and show how the resulting detector structures lead to considerable improvements in performance with respect to the correlation receiver, which has been widely considered in the literature and makes use of the Gaussian noise assumption. As a result of this work, analytical expressions for performance measures such as the probability of error in watermark decoding and probabilities of false alarm and detection in watermark detection are derived and contrasted with experimental results.

In [86] Jiwu et al have addressd a problem related to the embedding strategy for invisible image watermarking in the DCT domain. They have presented a quantitative analysis on the magnitudes of DCT components for a few commonly used images with different texture features It has been demonstrated that dc components should be used to place watermarks in order to improve robustness of watermarks. All

the previous methods avoided doing so. Based on the analysis, an embedding strategy that is different from the existing watermarking schemes has been proposed. As an application of the novel embedding strategy, an adaptive watermarking algorithm utilizing the feature of texture masking of HVS is presented The watermarks embedded with the proposed algorithm are invisible and very robust.

In[87] Shih et al have developed a general compressed domain image-watermarking framework based on zerotree-based coding. The binary watermark sequence is directly engraved into the quantizer's refinement outputs. By jointly considering watermarking

and compression, the proposed method provides superior robustness to conventional transform-domain approaches and reduces the systems complexity. Experimental results show that the propose method survives the attacks in the StirMark benchmark system.

The hierarchical decomposition structure of the wavelet transform has been widely used in image compression and other image processing researches [88]. Given its suitability to model the HVS behavior, the DWT has gained interest among watermarking researchers, as it is witnessed by the number of algorithms following this approach that have been proposed over the last few years. Some methods directly take inspiration from the most popular wavelet based compression algorithms.

In [8,90]the original image is required for watermark embedding. Xia *et al.* [90] add a pseudorandom sequence to the largest coefficients of the detail bands: perceptual considerations are taken into account by setting the amount of modification proportional to the strength of the coefficient itself. Watermark detection is achieved through comparison with the original un-watermarked image. It is also worth mentioning the work of Swanson *et al.* [96], where the DWT transform is applied to a video sequence by decomposing it, along the temporal axis, into stationary and moving components. Kundur *et al.* [89], for example, first decompose a binary logo through DWT, then repeatedly add it to the sub bands of the DWT decomposition of the host image; before being added, the watermark is scaled by a *salience* factor, computed on a block by block basis, related to the local image noise sensibility: visual masking is thus exploited up to a block resolution.

In [91] Rakesh et al have presented a method which adds the watermark to the significant coefficients in the DWT domain *and* does not require the original image in the detection process. Since the watermark is added to significant coefficients in the DWT domain, this method is much more resistant to common image manipulations. The method uses a three level DWT with a Daubechies 8-tap filter. The low pass sub-band is left out and all coefficients in the other sub-bands which are above a given threshold (TI) are picked. Watermark is added to these coefficients only. An *image sized* watermark has been used. Thus the watermark at a particular location in the DWT of the image is fixed; there is no dependence on the order of the significant coefficients (which can change due to image manipulations) in the detection process. Since watermark detection involves

finding the correlation coefficient, which is very sensitive to changes in the order of the vectors being correlated, order independence is a crucial factor in the success of the proposed method.

In [83](proposed by T. Hsu et al), the binary logo and the image are hierarchically decomposed (the image through DWT), each detail subband of the logo is then embedded into the corresponding detail subband of the image (more bits of the binary logo are embedded into more active image locations), the original image is required for watermark extraction. Image activity is estimated blockwise through variance computation.

In [94] D. Kudur et al have proposed a scheme where a binary code is embedded by suitably quantizing some of the coefficients of the detail bands: for watermark recovery the embedded binary code is estimated by analyzing coefficients quantization, once the code has been estimated it is correlated with the watermark and the result compared to a threshold chosen on the basis of a given false positive probability. No particular attention is given to visual masking.

In [98] H. Inoue et al and [92] H.-J. M.Wang et al have proposed schemes where the most significant DWT coefficients are selected and modified to carry the watermark. In the first case, some side information (i.e., the location of the modified coefficients) is required to recover the watermark. In the second, an algorithm is proposed for identifying *a posteriori* the modified coefficients. To take into account visual effects, in [101] very large coefficients are left unchanged, while in [92] the watermarking signal is weighted according to a band-dependent value.

In [97] (proposed by W. Zhu et al): each DWT coefficient of high pass bands is modified proportionally to its magnitude.

Nicchiotti *et al.* [95] choose to embed the watermark into the low pass band, by imposing a given difference among the mean values of two equally sized, randomly selected, subsets of the low pass image; the original image is not required for watermark detection. No particular care is taken with reference to perceptual masking.

In [99] Min-Jen et al have introduced an algorithm, which incorporates the wavelet transform and the spatial transform with the modular based threshold scheme for watermark embedding and extraction. The technique proposed has superior robustness of sensitive data protection under attacks like compression. There is no error up to 9:1

compression and as little as 5% error point up to 14:1 compression for extracted watermark. The watermark contain recognizable visual content which is meaningful in ownership identification. The classification and adjustment approach for the management of wavelet coefficients results in saving for side information communication.

In [100] Mauro Barni et al have proposed a scheme where in contrast to conventional methods operating in the wavelet domain, masking is accomplished pixel by pixel by taking into account the texture and the luminance content of all the image subbands. The watermark consists of a pseudorandom sequence which is adaptively added to the largest detail bands. As usual, the watermark is detected by computing the correlation between the watermarked coefficients and the watermarking code, anyway the detection threshold is chosen in such a way that the knowledge of the watermark energy used in the embedding phase is not needed, thus permitting to adapt it to the image at hand.

In [101] Xiangui Kang et al have proposed a blind discrete wavelet transform-discrete Fourier transform (DWT-DFT) composite image watermarking algorithm that is robust against both affine transformation and JPEG compression is proposed. This algorithm improves the robustness via using new embedding strategy, watermark structure, 2-D interleaving, and synchronization technique. A spread-spectrum-based informative watermark with a training sequence are embedded in the coefficients of the LL subband in the DWT domain while a template is embedded in the middle frequency components in the DFT domain. In watermark extraction, we first detect the template in a possibly corrupted watermarked image to obtain the parameters of affine transform and convert the image back to its original shape. Then we perform translation registration by using the training sequence embedded in the DWT domain and finally extract the informative watermark.

3.5 Advantages of Compressed Domain

This finds application for embedding watermark in video data. A real-time watermarking algorithm should meet several requirements. It should be a low complexity algorithm i.e. fully decompressing the video data, adding a watermark to the raw video data and finally compressing the data again is not an option for real time watermark embedding. The

watermark should be embedded and detected directly in the compressed stream to avoid computationally demanding operations.

3.6 Algorithms in the Compressed Domain

In real-time watermarking applications, robustness is not the only factor that plays an important role. The other factor that plays a very important role is computational complexity. In general, image or video data is transmitted in JPEG or MPEG compressed form. Real-time watermark embedding must take into account this compressed form, because first decompressing the data, adding a watermark and then recompressing the data is computationally too demanding. Therefore, it is desirable to develop watermarking techniques that can operate directly on the compressed bit stream, the code words, or the DCT transformed coefficients because then it is not necessary to fully decompress and recompress the data. In the VLC domain watermarking of MPEG-2 compressed data has been proposed by Langelaar et al [1] concerning a system, which is basically a LSB system. Chun Shein et al [2] proposed watermarking in the VLC domain for MPEG-2 Compressed data in the case of a fragile system. Mobasseri et al [3] also proposed watermarking for a robust system in the VLC domain for MPEG-2 compressed data.

In [119] a method is proposed that adds a DCT transformed pseudorandom pattern directly to the DC-DCT coefficients of an MPEG compressed video stream. The watermarking process only takes the luminance values of the I-frames into account. To embed a watermark the following procedure is performed:

- . First a pseudorandom pattern consisting of the integers $\{-1,1\}$ is generated based on a secret key. This pattern has the same dimensions as the I-frames.
- . Next, the pattern is modulated by a watermark bit string and multiplied by a gain factor.
- . Finally, the 8×8 block DCT transform is applied on the modulated pattern and the resulting DC-coefficients are added to the corresponding DC-values of each I-frame. The watermark can be detected using correlation techniques in the DCT domain or in the spatial domain as described earlier. The authors report that the algorithm decreases the visual quality of the video stream drastically. Therefore, the gain factor of the watermark has to be chosen to be very low (<1) and the number of pixels per watermark bit has to be

chosen to be extremely high ($\gg 100,000$) to maintain reasonable visual quality for the resulting video stream. This is mainly due to the fact that the watermark pattern is embedded in just one of the 64 DCT coefficients, the DC-component. Furthermore, the pattern consists only of low frequency components to which the human eye is quite sensitive.

In [120-123] and [124] a more sophisticated watermarking algorithm is proposed that embeds a watermark not only in the DC-coefficients, but also in the AC-coefficients of each I-, P-, and B-frame. The watermark here is also a pseudorandom pattern consisting of the integers $\{-1,1\}$ generated based on a secret key. This pattern has the same dimensions as the video frames. The pattern is modulated by a watermark bit string and multiplied by a gain factor k . To embed the watermark, the watermark pattern $W(x, y)$ is divided into 8×8 blocks. These blocks are transformed to the DCT domain and denoted by $W_{x,y}(u,v)$ where $x,y = 0,8,16,\dots$ and $u,v = 0,\dots,7$. Next, the 2-D blocks $W_{x,y}(u,v)$ are reordered in a zigzag scan fashion and become arrays $W_{x,y}(i)$, where $i = 0, 63, \dots$, $W_{x,y}(0)$ represents the DC-coefficient and $W_{x,y}(63)$ denotes the highest frequency AC coefficient of a 8×8 watermark block. Since the corresponding MPEG encoded 8×8 video content blocks are encoded in the same way as $I_{x,y}(i)$, these arrays can directly be used to add the watermark. For each video block $I_{x,y}(i)$ out of an I-,P-, or B-frame the following steps are performed:

1. The DC-Coefficient is modulated as follows:

$$I_{W_{x,y}}(0) = I_{x,y}(0) + W_{x,y}(0) \quad \text{Eqn.3.11}$$

which means that the average value of the watermark block is added to the average value of the video block.

2. To modulate the AC-coefficients the bit stream of the encoded video block is searched VLC-by-VLC for the next VLC code word, representing the next nonzero DCT coefficient. The run and level of this code word are decoded to determine its position i along the zigzag scan and its amplitude $I_{x,y}(i)$. A candidate DCT coefficient for the watermarked video block is generated, which is defined as

$$I_{W_{x,y}}(i) = I_{x,y}(i) + W_{x,y}(i), \quad i \neq 0 \quad \text{Eqn.3.12}$$

Now the constraint that the video bit rate may not be increased comes into play. The size S_{Z_I} of the VLC needed to encode $I_{x,y}(i)$ and the size $S_{Z_{I_w}}$ of the VLC needed to encode $I_{w_{x,y}}(i)$, are determined using the VLC-Tables B.14 and B.15 of the MPEG-2 standard [128]. If the size of VLC encoding the candidate DCT coefficient is equal or smaller than the size of the existing VLC, the existing VLC is replaced. Otherwise the VLC is left unaffected. This means that the DCT coefficient $I_{x,y}(i)$ is modulated in the following way:

$$\begin{aligned} \text{If } S_{Z_{I_w}} \leq S_{Z_I} \text{ then } I_{w_{x,y}}(i) &= I_{x,y}(i) + W_{x,y}(i) \\ \text{else } I_{w_{x,y}}(i) &= I_{x,y}(i) \end{aligned} \quad \text{Eqn.3.13}$$

This procedure is repeated until all AC-coefficients of the encoded video block are processed. To extract the watermark information, the MPEG encoded video stream is first fully decoded and the watermark bits are retrieved by correlating the decoded frames with the watermark pattern $W_{x,y}(i)$ in the spatial domain using the standard techniques. A major problem of directly modifying DCT-coefficients in an MPEG encoded video stream is drift or error accumulation. In an MPEG encoded video stream predictions from previous frames are used to reconstruct the actual frame, which itself may serve as a reference for future predictions. The degradation caused by the watermarking process may propagate in time and may even spatially spread. Since all video frames are watermarked, watermarks from previous frames and from the current frame may accumulate and result in visual artifacts. Therefore, a drift compensation signal Dr must be added. This signal must be equal to the difference of the (motion compensated) predictions from the unwatermarked bit stream and the watermarked bit stream.

Equation (3.12) changes for a drift compensated watermarking scheme into

$$I_{w_{x,y}}(i) = I_{x,y}(i) + W_{x,y}(i) + Dr_{x,y}(i) \quad \text{Eqn.3.14}$$

A disadvantage of this drift signal is that the complexity of the watermark embedding algorithm increases substantially, since an additional DCT operation and a complete MPEG decoding step are required to calculate the drift compensation signal. Due to the bit-rate constraint, only around 10-20% of the DCT coefficients are altered by the watermark embedding process, depending on the video content and the coarseness of the

MPEG quantizer. In some cases, especially for very low bit-rate video, only the DC-coefficients of the video stream are watermarked, the embedded watermark is video content dependent. In areas with only low-frequency content, the watermark can be embedded, typically around 0.5 ... 3% [124]. Since only existing (nonzero) DCT automatically consists of only low frequency components. This complies with the HVS. The watermark energy is mainly embedded in areas containing a lot of video content energy.

In a compressed bit stream we have direct access to the code words used in the compression algorithm. Watermark is embedded by modifying the VLC code words, yielding a computationally efficient watermarking method with a high payload [127], [126]. The technique is described as follows. A watermark consisting of l label bits $b_j (j = 0, 1, 2, \dots, l-1)$ is embedded in the MPEG-stream by selecting suitable VLC and forcing the LSB of their *quantized* level to the value of b_j . To ensure that the change in the VLC is perceptually invisible after decoding and that the MPEG-bit stream keeps its original size, only those VLC for which another VLC exists with:

- the same run length,
- a quantized level difference of one,
- the same code word length.

are chosen.

Table 3.1
VLC in Table
MPEG-2 Standard

Variable Length Code	VLC	Run	Level	LSB of Level
0010 0110 s	8+1	0	5	1
0010 0001 s	8+1	0	6	0
0000 0001 1101 s	12+1	0	8	0
00000 0001 1000 s	12+1	0	9	1
0000 0000 1101 0s	13+1	0	8	0
0000 0000 1100 1s	13+1	0	9	1
0000 0000 0111 11s	14+1	0	16	0
0000 0000 0111 10 s	14+1	0	17	1
0000 0000 0011 101 s	15+1	1	10	0
0000 0000 0011 100 s	15+1	1	11	1
0000 0000 0001 0011 s	16+1	1	15	1
0000 0000 0001 0010 s	16+1	1	16	0

Example of lc-
B.14 of the

A VLC that meets this requirement is called a label-bit-carrying-VLC (lc-VLC). According to Tables B.14 and B.15 of the MPEG-2 standard [47], an abundance of such lc-VLCs exists. Furthermore, all fixed-length-coded DCT-coefficients following an Escape- code meet the requirement. Some examples of lc-VLCs are listed in Table 3.1, where the symbol s represents the sign-bit. This sign-bit represents the sign of the DCT coefficient level. The VLC in the intra- and inter coded macro blocks can be used in the watermarking process. The DC coefficients are not used, because they are predicted from other DC coefficients and coded with a different set of VLC and Escape-codes. Furthermore, replacing each DC coefficient in intra- and inter coded frames can result in visible artifacts due to drift. By only taking the AC coefficients into account the watermark will adapt itself more to the video content and the drift will be limited. To add the label bit stream L to an MPEG-video bit stream, the VLC in each macro block are tested. If an lc-VLC is found and the LSB of its level is unequal to the label bit b_j ($j = 0, 1, 2, \dots, l-1$) this VLC is replaced by another one, whose LSB-level represents the label bit. If the LSB of its level equals the label bit b_j the VLC is not changed. The procedure is repeated until all label bits are embedded. In Fig.3.1 an example is given of the watermarking process, where three label bits are embedded in the MPEG video stream. To extract the label bit stream L the VLCs in each macro blocks are tested. If an lc-VLC is found, the value represented by its LSB is assigned to the label bit b_j . The procedure is repeated for $j = 0, 1, 2, \dots, l-1$ until no lc-VLCs can be found anymore. This technique gives a high payload (up to 29 kbit/s) without significant perceptible quality degradation [65]. The watermark embedded with this method can easily be removed by decoding and reencoding the video stream or by relabeling the stream using another randomly generated watermark pattern.

This technique has been extended to make it resistant to relabeling [65], as follows. The watermark label bits b_i are not directly stored in the LSB of the VLC, but a one-dimensional pseudorandom watermark pattern $W(x)$ is generated consisting of the integers $\{-1, 1\}$ based on a secret key, which is modulated with the label bits b_i . To add

this modulated pattern to the video stream, only those VLC for which two other VLC exist, with the same run length and the same codeword length are selected. One VLC must have a level difference of $+\delta$ and the other VLC must have a level

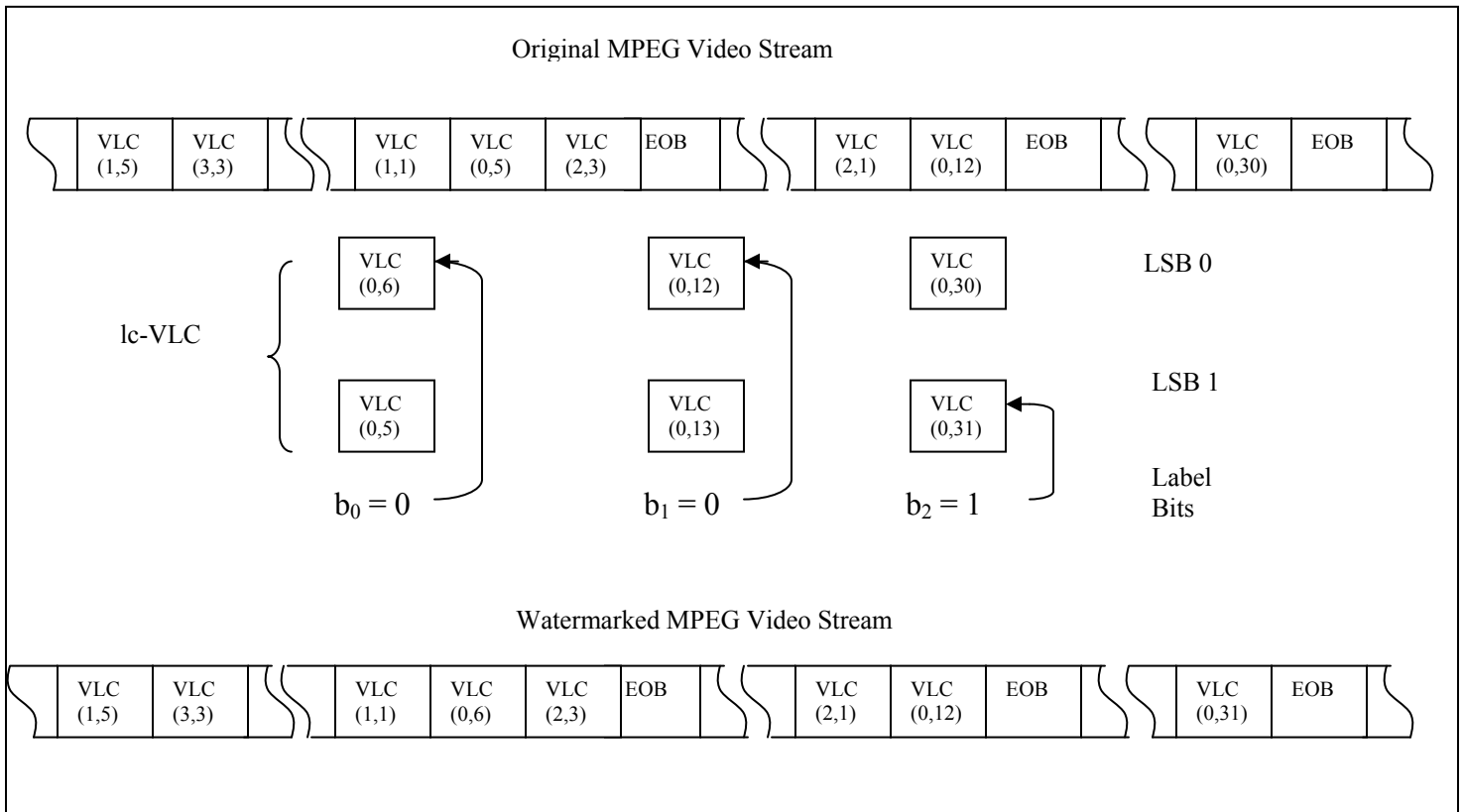
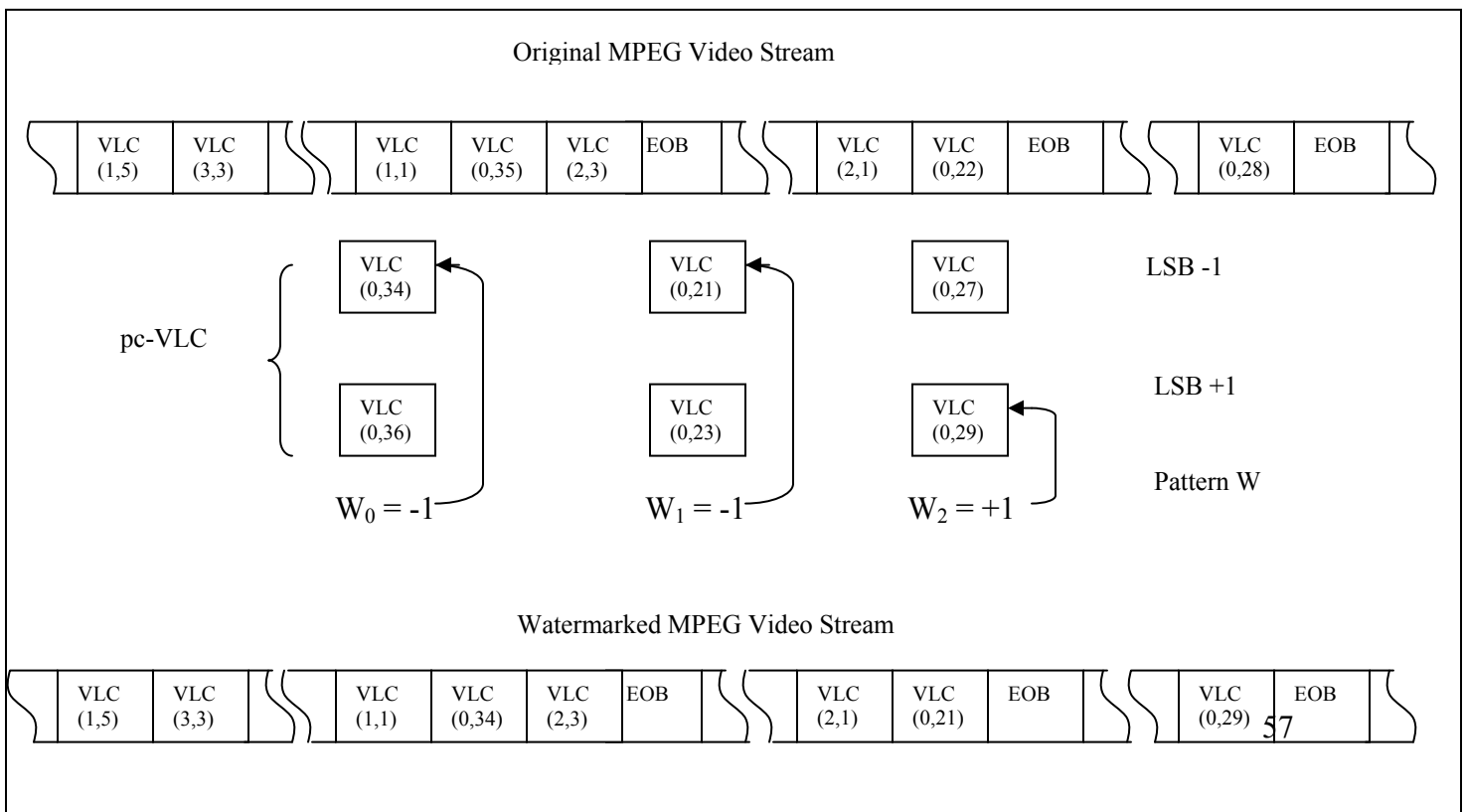


Figure 3.1 Example of the LSB Watermarking Process. The (x.y) pairs represent the (zero run, level) pairs used in the MPEG VLC Encoding



difference of $-\delta$. Most lc-VLC meet these requirements for a relative small δ (e.g., $\delta=1,2,3$). For notational simplicity these are called pattern-carrying-VLC (*pc-VLC*). To embed a watermark in a video stream, the modulated watermark pattern are added to the levels of the *pc-VLC*. To extract the watermark, the *pc-VLC* are collected in an array. The watermark label bits can now be retrieved by calculating the correlation between this array of *pc-VLC* and the secret watermark pattern $W(x)$. In Fig. 3.2 an example is given of the watermark embedding process. About 1,000,...,10,000 *pc-VLC* are required to encode one watermark label bit b_i and thus drastically reduce the payload of the watermark. However, several watermark label bit strings can be added without interfering with each other, if independent pseudorandom patterns are used to form the basic pattern $W(x)$.

3.7 Conclusion

In contrast to the spatial-domain-based watermarking, frequency-domain-based techniques can embed more bits of watermark and are more robust to attack. Online application of watermarking for video in the spatial domain becomes cumbersome due to associated high computational complexities involved. On the other hand, Watermarking in the DCT domain needs preprocessing operations such as inverse entropy coding and inverse quantization [129]. Watermarking of compressed data in the VLC domain, the computational complexities involved in the spatial domain and preprocessing operations involved in the DCT domain can be spared with.

CHAPTER 4

REVERSIBLE WATERMARKING IN THE VLC DOMAIN of MPEG-2

OBJECTIVE: To obtain a Reversible Watermarking Scheme in the VLC domain of MPEG-2, to obtain online watermarking scheme

CHAPTER ORGANISATION:

- 4.1 Insight into Reversible Watermarking
- 4.2 Requirement of online application
- 4.3 Insight into MPEG-2 Compression
- 4.4 The proposed Algorithm
- 4.5 Results
- 4.6 Conclusion

4.1 Insight into Reversible Watermarking

Reversible watermark is a special subset of fragile watermark. Like all fragile watermarks, it can be used for digital content authentication. But reversible watermark is much more than content authentication. It has an additional advantage that when watermarked content has been detected to be authentic, one can remove the watermark to retrieve the original un-watermarked content.

Reversible watermarking can be classified into SSR (strict sense reversible) and WSR (wide sense reversible). A watermark is SSR if once it has been decoded/detected it can also be removed from the host asset, thus making it possible the exact recovery of the original asset. A watermark is WSR if once it has been decoded /detected it can be made undecodable/undetectable without producing any perceptible distortion of the host asset..

In reversible watermarking, a watermark is embedded in a digital image I . This results in a watermarked image I' . This image might or might not have been tampered by some intentional or unintentional attack. The watermark can be removed from I' to restore the original image, which results in a new image I'' ([provided no tampering has taken place]). By definition of Reversible watermark the restored image I'' will be exactly same as the original image I , pixel-by-pixel, bit-by-bit. Fig. 4.1 illustrates Reversible Watermarking Scheme.

The motivation of reversible watermark is distortion-free embedding. One basic requirement of digital watermarking is its imperceptibility, embedding a watermark inevitably changes the original content. Even a very slight change in pixel values may not be desirable in sensitive imagery, such as military data, medical data and data used in crime detection. In such scenario, every bit of information is important. Any change will affect the intelligence of the digital content, and the access to the original, raw data is always required. Reversible watermarks will provide the original, raw data for digital content authentication.

A basic approach of reversible watermarking algorithms [130 - 141] is to select an embedding area in an image, and embed both the payload and the original values in this area. As the amount of information needed to be embedded (payload and original values in the original area) is larger than that of the embedding area,

most reversible watermarking techniques [130- 133] rely on loss less data compression on the original values in the embedding area, and the space saved from compression is used for embedding the payload. Recently value expansion [134-141] techniques are more and more popular as a high performance (embedding capacity vs. distortion) means for reversible data embedding. These value expansion based algorithms are stretching the performance of reversible watermarking to the optimal case defined by information theory. There are two types of value expansion schemes proposed till now:

1. Value addition

Value addition [134,137,138] adds a constant T (≥ 1) to some selected scale values in a histogram to evaluate a range of space in the histogram for embedding. Value addition was first used for reversible watermarking by [134] and adopted in integer DCT domain by [137], which select the peak–amplitude scale value P (with absolute amplitude L) in the histogram to do the value addition as shown in equation 4.1.

$$\text{if } (x > P) \quad x' = x + 1 \quad \text{else} \quad x' = x \quad \text{Eqn. 4.1}$$

In equation 4.1 x are the original scale values and x' are the resulting ones after value addition. This value addition process is equivalent to shifting part of the histogram ($x > P$) with 1 unit to the right side and the original scale value $x = P + 1$ is evaluated for watermarking. The embedding rule is to distribute the L original P -valued elements $el(j)$ ($j = 1, 2, \dots, L$) to the scale value P or $P + 1$ as a modulation of watermark bits $wm(j)$ in accordance with equations 4.2 and 4.33.

$$el(j) = P \quad \text{Eqn.4.2}$$

$$\text{if } wm(j) = 0 \quad (j = 1, 2, \dots, L)$$

$$el(j) = P + 1 \quad \text{Eqn.4.3}$$

$$\text{if } wm(j) = 1 \quad (j = 1, 2, \dots, L)$$

If N scale values P_1, P_2, \dots, P_N are selected to do the value addition one by one from left to right, the new positions of the selected scale values will be $P_1, P_2 + 1, \dots, P_N + N - 1$ and the original N scale values $P_1 + 1, P_2 + 2, \dots, P_N + N$ will be

evacuated for watermarking. This value addition process is known as the Wedge mode.

For an example:

Let $N = 3$, the range of scale values be from 0 to 9, and $L1$, $L2$ and $L3$ be the absolute amplitudes of $P_1 = 2$, $P_2 = 5$, $P_3 = 8$ respectively. The embedding capacity is $L1+L2+L3$. The new positions are given by P_1 , P_2+1 , P_3+2 i.e. 2, 6, 10. So the bits are modulated to the scale values 2, 3, 6 and 7 according to equations 4.2. And 4.3.

Another value addition process for the N watermark modulation scale values P_1, P_2, \dots, P_N is to maintain the original scale values from 0 to P_N in their original positions, but evacuate the N scale values from $P_N + 1$ to $P_N + N$ for watermarking. This is known as rain-check mode.

For an example:

Let $N = 3$, the range of scale values be from 0 to 9, and $L1$, $L2$ and $L3$ be the absolute amplitudes of $P_1 = 2$, $P_2 = 5$, $P_3 = 8$ respectively. The embedding capacity is $L1+L2+L3$. The new positions are given by P_1 , P_2 , P_3 i.e. 2, 5, 8. So the bits are modulated to the scale values 2, 5, 8, 9, 10 and 11 according to equations 4.2 and 4.3.

The algorithms in [134] and [137] are just the 1-peak-amplitude case of value addition performed in spatial and integer DCT domain respectively.

A special case of the value addition scheme is to select the former N scale values

$P_i = I - 1$ ($I = 1, 2, \dots, N$) in a histogram for value addition, and the original scale values $x \geq P_N + 1$ is added by N to $x' \geq P_N + N + 1$. In this special case, the scale values $0 \sim P_N$ can be modulated in $[0, P_N + N]$ using any value addition scheme: the Wedge mode or Rain-Check mode or any other mode. In the spread spectrum reversible-watermarking scheme proposed in [138], an offset A is used to modulate the watermark in the range $[-A+1, A-1]$.

2. Bit-shifting

Bit shifting [135,136,139,140] multiplies some selected values by 2 and evaluates the LSB for embedding. Particularly for the bit-shifting operation,

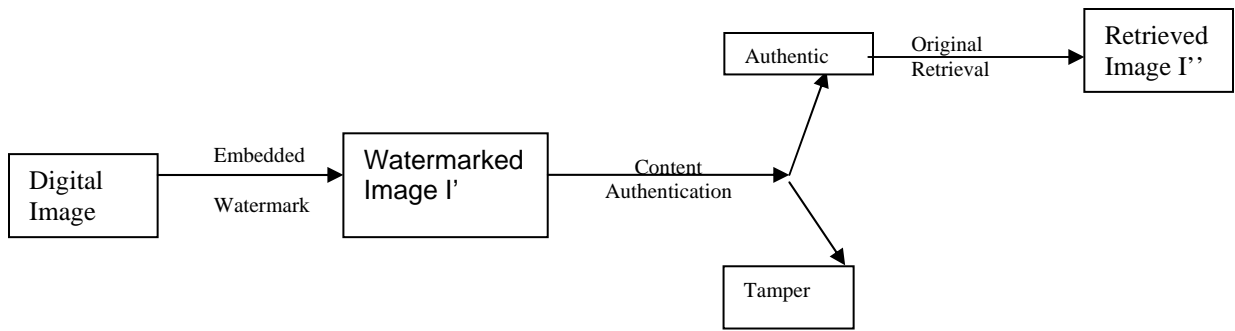


Figure 4.1 A Reversible Watermarking Scheme

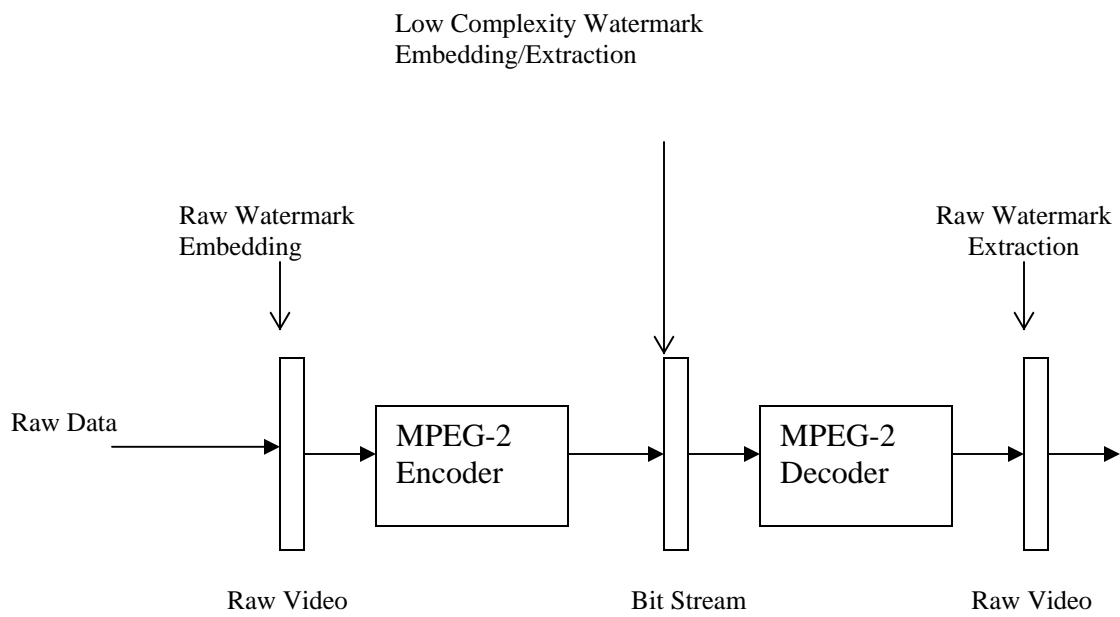


Figure 4.2. Watermark Embedding/Extraction in Raw vs. Compressed Video

small amplitude values are preferred for expansion because the distortion caused by bit shifting will be accordingly slight, and therefore, high frequency coefficients in the transformed domains [135,136,139] and prediction error [140] are suitable and used for the bit shifting operation. The main idea of bit-shifting is to embed watermark bit '0' or '1' in the LSB evacuated by left-shifting operation (equal to multiplied by 2) on signal's original amplitudes. The difference expansion scheme [pap1, pap2] uses it in integer transform in spatial domain. The difference expansion [135] and companding scheme [139] uses it in the integer wavelet domain, [136] uses it in the integer wavelet domain and [140] uses it in the prediction-error domain.

Watermark reversibility must be carefully considered when robustness/security of the hidden information is a major concern, since it implies that only trusted users are allowed to read/detect the watermark, thus complicating considerably the design of suitable application protocols. Reversible watermarking finds application in crime detection and medical data authentication

4.2 Requirement of Online Application

A real-time watermarking algorithm should meet several requirements. In the first place it should be an oblivious low complexity algorithm. This means that fully decompressing the video data, adding a watermark to the raw data and finally compressing the data again is not an option for real-time watermark embedding. The watermark should be embedded and detected in the compressed stream to avoid computationally demanding operations (Fig.4.2) The watermark embedding operation should not increase the size of the compressed video stream. If size of the stream increases, transmission over a fixed bit rate channel would cause problems: the buffers in hardware decoders can run out of space, or the synchronization of audio and video can be disturbed.

4.3 Insight into MPEG-2

4.3.1 Introduction

MPEG-2 is an extension of the MPEG-1 international standard for digital compression of audio and video signals. MPEG-1 was designed to code progressively scanned video at bit rates up to about 1.5 Mbit/s for applications such as CD-i (compact disc interactive). MPEG-2 is directed at broadcast formats at higher data rates; it provides extra algorithmic 'tools' for efficiently coding interlaced video, supports a wide range of bit rates and provides for multichannel surround sound coding. Recent progress in digital technology has made the widespread use of compressed digital video signals practical. Standardization has been very important in the development of common compression methods to be used in the new services and products that are now possible. This allows the new services to interoperate with each other and encourages the investment needed in integrated circuits to make the technology cheap.

MPEG (Moving Picture Experts Group) was started in 1988 as a working group within ISO/IEC with the aim of defining standards for digital compression of audio-visual signals. MPEG's first project, MPEG-1, was published in 1993 as ISO/IEC 11172 [143]. It is a three-part standard defining audio and video compression coding methods and a multiplexing system for interleaving audio and video data so that they can be played back together. MPEG-1 principally supports video coding up to about 1.5 M bit/s giving quality similar to VHS and stereo audio at 192 bit/s. It is used in the CD-i and Video-CD systems for storing video and audio on CD-ROM. During 1990, MPEG recognized the need for a second, related standard for coding video for broadcast formats at higher data rates. The MPEG-2 standard [142] is capable of coding standard-definition television at bit rates from about 3-15 M bit/s and high-definition television at 15-30 M bit/s. MPEG-2 extends the stereo audio capabilities of MPEG-1 to multi-channel surround sound coding. MPEG-2 decoders will also decode MPEG-1 bit streams. MPEG-2 [144] aims to be a *generic* video coding system supporting a diverse range of applications. Different algorithmic 'tools', developed for many applications, have been integrated into the full standard. To implement all the features of the standard in all decoders is unnecessarily complex and a waste of bandwidth, so a small number of subsets of the full standard, known as profiles and levels, have been defined. A profile is a subset of algorithmic tools

and a level identifies a set of constraints on parameter values (such as picture size and bit rate). A decoder, which supports a particular profile and level, is only required to support the corresponding subset of the full standard and set of parameter constraints.

4.3.2 Video Fundamentals

Television services in Europe currently broadcast video at a frame rate of 25 Hz. Each frame consists of two interlaced fields, giving a field rate of 50 Hz. The first field of each frame contains only the odd numbered lines of the frame (numbering the top frame line as line 1). The second field contains only the even numbered lines of the frame and is sampled in the video camera 20 ms after the first field. It is important to note that one interlaced frame contains fields from two instants in time. American television is similarly interlaced but with a frame rate of just under 30 Hz. In video systems other than television, non-interlaced video is commonplace (for example, most computers output non-interlaced video). In non-interlaced video, all the lines of a frame are sampled at the same instant in time. Non-interlaced video is also termed 'progressively scanned' or 'sequentially scanned' video. The red, green and blue (RGB) signals coming from a colour television camera can be equivalently expressed as luminance (Y) and chrominance (UV) components. The chrominance bandwidth may be reduced relative to the luminance without significantly affecting the picture quality. For standard definition video, CCIR recommendation 601 [145] defines how the component (YUV) video signals can be sampled and digitized to form discrete *pixels*. The terms 4:2:2 and 4:2:0 are often used to describe the sampling structure of the digital picture. 4:2:2 means the chrominance is horizontally sub sampled by a factor of two relative to the luminance; 4:2:0 means the chrominance is horizontally and vertically sub sampled by a factor of two relative to the luminance. The active region of a digital television frame, sampled according to CCIR recommendation 601, is 720 pixels by 576 lines for a frame rate of 25 Hz. Using 8 bits for each Y, U or V pixel, the uncompressed bit rates for 4:2:2 and 4:2:0 signals are therefore:

$$4:2:2: 720 \times 576 \times 25 \times 8 + 360 \times 576 \times 25 \times (8 + 8) = 166 \text{ M bit/s}$$

$$4:2:0: 720 \times 576 \times 25 \times 8 + 360 \times 288 \times 25 \times (8 + 8) = 124 \text{ M bit/s}$$

MPEG-2 is capable of compressing the bit rate of standard-definition 4:2:0 video down to about 3-15 M bit/s. At the lower bit rates in this range, the impairments introduced by the MPEG-2 coding and decoding process become increasingly objectionable. For digital terrestrial television broadcasting of standard-definition video, a bit rate of around 6 M bit/s is thought to be a good compromise between picture quality and transmission bandwidth efficiency.

4.3.4 Bit Rate Reduction Principle

A bit rate reduction system operates by removing redundant information from the signal at the coder prior to transmission and re-inserting it at the decoder. A coder and decoder pair is referred to as a 'codec'. In video signals, two distinct kinds of redundancy can be identified.

4.3.4.1 Spatial and Temporal Redundancy:

Pixel values are correlated with their neighbours both within the same frame and across frames. Therefore, to some extent, the value of a pixel is predictable given the values of neighbouring pixels.

4.3.4.2 Psychovisual Redundancy:

The human eye has a limited response to fine spatial detail [146], and is less sensitive to detail near object edges or around shot-changes. Consequently, controlled impairments introduced into the decoded picture by the bit rate reduction process should not be visible to a human observer.

Two key techniques employed in an MPEG codec are intra-frame Discrete Cosine Transform (DCT) coding and motion-compensated inter-frame prediction. These techniques have been successfully applied to video bit rate reduction prior to MPEG, notably for 625-line video contribution standards at 34 M bit/s [147] and video conference systems at bit rates below 2 M bit/s [148].

4.3.5 Intra-frame DCT Coding

DCT [149]: A two-dimensional DCT is performed on small blocks (8 pixels by 8 lines) of each component of the picture to produce blocks of DCT coefficients (Fig. 1). The magnitude of each DCT coefficient indicates the contribution of a particular combination

of horizontal and vertical spatial frequencies to the original picture block. The coefficient corresponding to zero horizontal and vertical frequency is called the DC coefficient.

The NXN two-dimensional DCT is defined as:

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad \text{Eqn.4.4}$$

$$C(u), C(v) = \frac{1}{\sqrt{2}} \quad \text{for } u, v = 0 \\ = 1 \text{ otherwise}$$

The inverse DCT or IDCT is defined as :

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v) F(u, v) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad \text{Eqn.4.2}$$

Here x, y are spatial coordinates in the image block

The DCT doesn't directly reduce the number of bits required to represent the block. In fact for an 8x8 block of 8 bit pixels, the DCT produces an 8x8 block of 11 bit coefficients (the range of coefficient values is larger than the range of pixel values.) The reduction in the number of bits follows from the observation that, for typical blocks from natural images, the distribution of coefficients is non-uniform. The transform tends to concentrate the energy into the low-frequency coefficients and many of the other coefficients are near zero. The bit rate reduction is achieved by not transmitting the near-zero coefficients and by quantising and coding the remaining coefficients as described below. The non-uniform coefficient distribution is a result of the spatial redundancy present in the original image block.

4.3.6 Quantization: The function of the coder is to transmit the DCT block to the decoder, in a bit rate efficient manner, so that it can perform the inverse transform to reconstruct the image. It has been observed that the numerical precision of the DCT coefficients may be reduced while still maintaining good image quality at the decoder. Quantization is used to reduce the number of possible values to be transmitted, reducing the required number of bits. The HVS Quantization matrix used in this work is given in (Fig. 4.2) The degree of quantization applied to each coefficient is weighted according to the visibility of the resulting quantization noise to a human observer. In practice, this

results in the high-frequency coefficients being more coarsely quantized than the low-frequency coefficients. The quantization noise introduced by the coder is not reversible in the decoder, making the coding and decoding process 'lossy'.

4.3.7 Coding: The serialization and coding of the quantized DCT coefficients exploits the likely clustering of energy into the low-frequency coefficients and the frequent occurrence of zero-value coefficients. The block is scanned in a diagonal zigzag pattern starting at the DC coefficient to produce a list of quantized coefficient values, ordered according to the scan pattern. The list of values produced by scanning is stored as a run level pair known as a tuple. This tuple is entropy coded using a variable-length code (VLC). Each VLC code word denotes a run of zeros followed by a non-zero coefficient of a particular level. VLC coding recognizes that short runs of zeros are more likely than long ones and small coefficients are more likely than large ones. The VLC allocates code words, which have different lengths depending upon the probability with which they are expected to occur. To enable the decoder to distinguish where one code ends and the next begins, the VLC has the property that no complete code is a prefix of any other. (Fig. 4.5) shows the zigzag scanning process, using the scan pattern common to both MPEG-1 and MPEG-2. MPEG-2 has an additional 'alternate' scan pattern intended for scanning the quantized coefficients resulting from interlaced source pictures. To illustrate the variable-length coding process, consider the following example list of values produced by scanning the quantized coefficients from the block represented in the transformed block (Fig. 4.5):

12, 6, 6, 0, 4, 3, 0, 0, 0...0 The first step is to group the values into runs of (zero or more) zeros followed by a non-zero value this is known as tuple code. Additionally, the final run of zeros is replaced with an end of block (EOB) marker. Using parentheses to show the groups, this gives:

(0,12), (0,6), (0,6), (1, 4), (0,3) EOB

The next step is to generate the variable length code words corresponding to each group (a run of zeros followed by a non-zero value) and the EOB marker. Table 4.1 shows an extract of the DCT coefficient VLC table (the complete VLC Tables B.15 and B.16 of

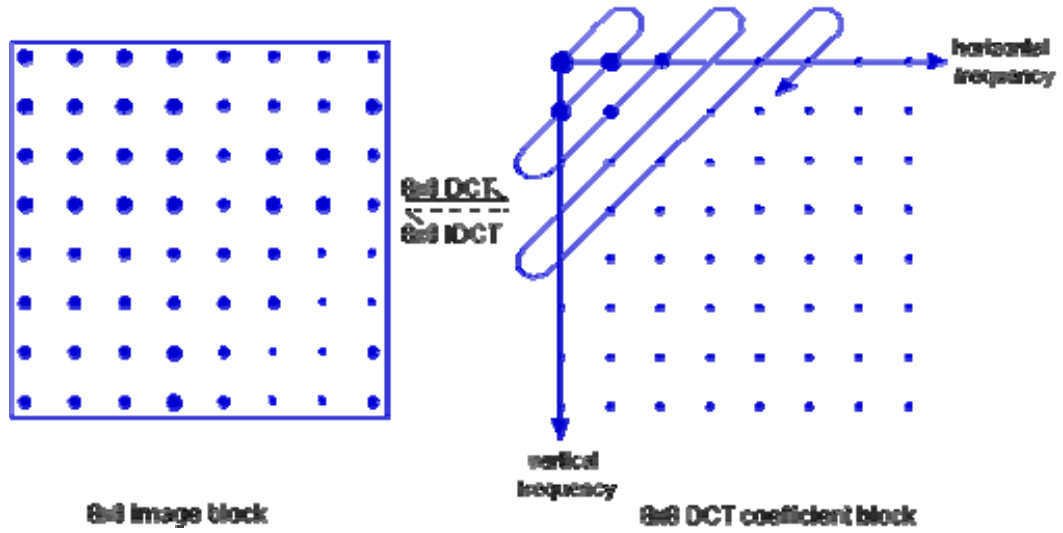


Figure 4.3: The discrete cosine Transform (DCT) Pixel value and DCT coefficient magnitude are represented by dot size

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Figure 4.4 Matrix Representing HVS Mask

Table 4.1 Extract from the MPEG-2 DCT Coefficient VLC Table

Length of run of zeros	Value of non-zero coefficient	Variable-length codeword
0	12	0000 0000 1101 00
0	6	0010 0001 0
1	4	0000 0011 000
0	3	0010 10
EOB	-	10

12	6	3	0	0	0	0	0
6	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 4.5 Zigzag Scan

ISO/ISE 96 standard is provided as tables 4.2 and 4.3. Using the variable length code from Table 1 and adding spaces and commas for readability, the final coded representation of the example block is:

0000 0000 1101 00, 0010 0001 0, 0010 0001 0, 0000 0011 000, 0010 10, 10

Table 4.2 (Table B.15 – DCT Coefficients Table 1)

Variable Length code (Note 1)	Run	Level
0110 (Note 2)	EOB	
10s	0	1
010s	1	1
110s	0	2
0010 1s	2	1
0 111s	0	3
0011 1s	3	1
0001 10s	4	1
0011 0s	1	2
0001 11s	5	1
0000 110s	6	1
0000 100s	7	1
1110 0s	0	4
0000 111s	2	2
0000 101s	8	1
1111 000s	9	1
0000 01	ESCAPE	
1110 1s	0	5
0001 01s	0	6
1111 001s	1	3
0010 0110s	3	2

1111 010S	10	1
0010 0001s	11	1
0010 0101s	12	1
0010 0100s	13	1
0001 00s	0	7
0010 011s	1	4
1111 11 00s	2	3
1111 11 01s	4	2
0000 0010 0s	5	2
0000 000 1s	14	1
0000 00 111s	15	1
0000 0011 01s	16	1
1111 011s	0	8
1111 100s	0	9
0010 0011s	0	10
0010 0010s	0	11
0010 0000s	1	5
0000 00 1100s	2	4
0000 0001 1100s	3	3
0000 0001 0010s	4	3
0000 0001 1110s	6	2
0000 0001 0101s	7	2
0000 0001 0001s	8	2
0000 0001 1111s	17	1
0000 0001 1010s	18	1
0000 0001 1001s	19	1
0000 0001 0111s	20	1
0000 0001 0110s	21	1
1111 1010s	0	12
1111 1011s	0	13

1111 1110s	0	14
1111 1111s	0	15
0000 0000 10110s	1	6
0000 0000 10101s	1	7
0000 0000 10100s	2	5
0000 0000 1001 1s	3	4
0000 0000 1001 0s	5	3
0000 0000 1000 1s	9	2
0000 0000 1000 0s	10	2
0000 0000 1111 1s	22	1
0000 0000 1111 0s	23	1
0000 0000 1110 1s	24	1
0000 0000 1110 0s	25	1
0000 0000 1101 1s	26	1
0000 0000 0111 11s	0	16
0000 0000 0111 10s	0	17
0000 0000 0111 01s	0	18
0000 0000 0111 00s	0	19
0000 0000 0110 11s	0	
	0	21
0000 0000 0110 01S	0	22
0000 0000 0110 00s	0	23
0000 0000 0101 11s	0	24
0000 0000 0101 10s	0	25
0000 0000 0101 01s	0	26
0000 0000 0101 00s	0	27
0000 0000 0100 11s	0	28
0000 0000 0100 10s	0	29
0000 0000 0100 01s	0	30
0000 0000 0100 00s	0	31

0000 0000 0011 000s	0	32
0000 0000 0010 111s	0	33
0000 0000 0010 110s	0	34
0000 0000 0010 101s	0	35
0000 0000 0010 100s	0	36
0000 0000 0010 011s	0	37
0000 0000 0010 010s	0	38
0000 0000 0010 001s	0	39
0000 0000 0010 000s	0	40
0000 0000 0011 111s	1	8
0000 0000 0011 110s	1	9
0000 0000 0011 101s	1	10
0000 0000 0011 100s	1	11
0000 0000 0011 011s	1	12
0000 0000 0011 010s	1	13
0000 0000 0011 001s	1	14
0000 0000 0001 0011s	1	15
0000 0000 0001 0010s	1	16
0000 0000 0001 0001s	1	17
0000 0000 0001 0000s	1	18
0000 0000 0001 0100s	6	3
0000 0000 0001 1010s	11	2
0000 0000 0001 1001s	12	2
0000 0000 0001 1000s	13	2
0000 0000 0001 0111s	14	2
0000 0000 0001 0110s	15	2
0000 0000 0001 0101s	16	2
0000 0000 0001 1111s	27	1
0000 0000 0001 1110s	28	1
0000 0000 0001 1101s	29	1

0000 0000 0001 1100s	30	1
0000 0000 0001 1011s	31	1
NOTES		
1. The Last bit s denotes the sign of the level 2. “EOB” shall not be the only code of the block		

Table 4.3 (Table B16 ISO/ISE 1996) - Encoding of Run and Level Following an Escape Code

Fixed Length code	Run	Fixed Length Code	Signed_level
0000 00	0	1000 0000 0001	-2047
0000 01	1	1000 0000 0010	-2046
0000 10	2
...	...	1111 1111 1111	-1
...	...	0000 0000 0000	Forbidden
...	...	0000 0000 0001	+1
...
1111 11	63	0111 1111 1111	+2047

4.3.8 Motion-Compensated Inter-Frame Prediction

This technique exploits temporal redundancy by attempting to predict the frame to be coded from a previous 'reference' frame. The prediction cannot be based on a source picture because the prediction has to be repeatable in the decoder, where the source pictures are not available (the decoded pictures are not identical to the source pictures because the bit rate reduction process introduces small distortions into the decoded picture.) Consequently, the coder contains a local decoder, which reconstructs pictures exactly as they would be in the decoder, from which predictions can be formed.

The simplest inter-frame prediction of the block being coded is that which takes the co-sited (i.e. the same spatial position) block from the reference picture. This makes a good

prediction for stationary regions of the image, but is poor in moving areas. A more sophisticated method, known as motion-compensated inter-frame prediction, is to offset any translational motion, which has occurred between the block being coded and the reference frame, and to use a shifted block from the reference frame as the prediction. One method of determining the motion that has occurred between the block being coded and the reference frame is a 'block-matching' search in which a large number of trial offsets are tested by the coder using the luminance component of the picture. The 'best' offset is selected based on minimum error between the block being coded and the prediction. The bit rate overhead of using motion-compensated prediction is the need to convey the motion vectors required to predict each block to the decoder. For example, using MPEG-2 to compress standard-definition video to 6 Mbit/s, the motion vector overhead could account for about 2 Mbit/s during a picture making heavy use of motion-compensated prediction.

4.3.9 MPEG-2 Details

4.3.9.1 Codec structure

In an MPEG-2 system, the DCT and motion-compensated inter frame prediction are combined, as shown in Fig. 2. The coder subtracts the motion-compensated prediction from the source picture to form a 'prediction error' picture. The prediction error is transformed with the DCT, the coefficients are quantized and these quantized values coded using a VLC. The coded luminance and chrominance prediction error is combined with 'side information' required by the decoder, such as motion vectors and synchronizing information, and formed into a bit stream for transmission. Fig. 3 shows an outline of the MPEG-2 video bit stream structure. In the decoder, the quantized DCT coefficients are reconstructed and inverse transformed to produce the prediction error. This is added to the motion-compensated prediction generated from previously decoded pictures to produce the decoded output. In an MPEG-2 codec, the motion-compensated predictor shown in Fig. 2 supports many methods for generating a prediction. For example, the block may be 'forward predicted' from a previous picture, 'backward predicted' from a future picture, or 'bidirectionally predicted' by averaging a forward and backward prediction. The method used to predict the block may change from one block to the next. Additionally, the two fields within a block may be predicted separately with their own

motion vector, or together using a common motion vector. Another option is to make a zero-value prediction, such that the source image block rather than the prediction error block is DCT coded. For each block to be coded, the coder chooses between these prediction modes, trying to maximize the decoded picture quality within the constraints of the bit rate. The choice of prediction mode is transmitted to the decoder, with the prediction error, so that it may regenerate the correct prediction.

4.3.9.2 Picture Types

In MPEG-2, three 'picture types' are defined. The picture type defines which prediction modes may be used to code each block. 'Intra' pictures (I-pictures) are coded without reference to other pictures. Moderate compression is achieved by reducing spatial redundancy, but not temporal redundancy. They can be used periodically to provide access points in the bit stream where decoding can begin. 'Predictive' pictures (P-pictures) can use the previous I- or P-picture for motion compensation and may be used as a reference for further prediction. Each block in a P-picture can either be predicted or intra-coded. By reducing spatial and temporal redundancy, P-pictures offer increased compression compared to I-pictures. 'Bidirectionally-predictive' pictures (B-pictures) can use the previous and next I- or P-pictures for motion-compensation, and offer the highest degree of compression. Each block in a B-picture, can be forward, backward or bidirectionally predicted or intra-coded. To enable backward prediction from a future frame, the coder reorders the pictures from natural 'display' order to 'bit stream' order so that the B-picture is transmitted after the previous and next pictures it references.

This introduces a reordering delay dependent on the number of consecutive B-pictures. The different picture types typically occur in a repeating sequence, termed a 'Group of Pictures' or GOP. A typical GOP in display order is:

B₁ B₂ I₃ B₄ B₅ P₆ B₇ B₈ P₉ B₁₀ B₁₁ P₁₂

The corresponding bit stream order is:

I₃ B₁ B₂ P₆ B₄ B₅ P₉ B₇ B₈ P₁₂ B₁₀ B₁₁

A regular GOP structure can be described with two parameters: N , which is the number of pictures in the GOP, and M , which is the spacing of P-pictures. The GOP given here is

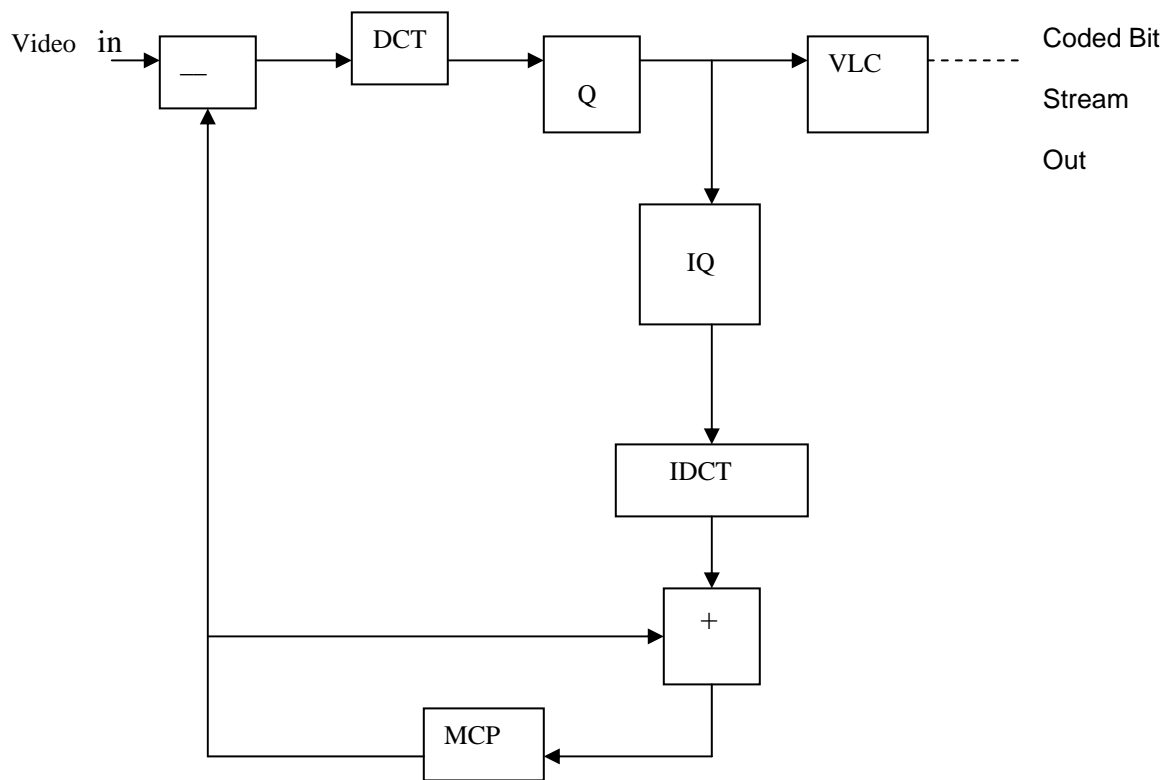


Figure 4.6 a) Motion Compensated DCT coder

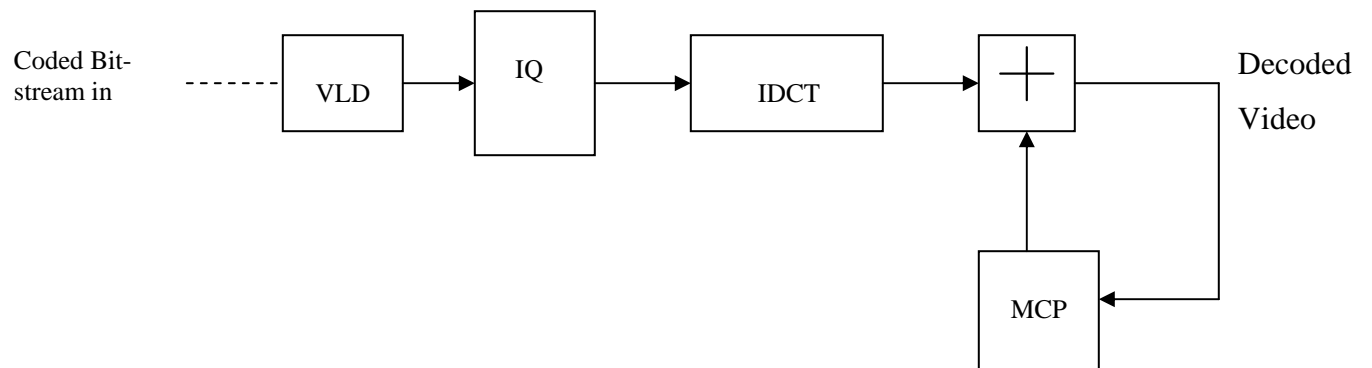


Figure 4.6 b) Motion Compensated DCT decoder

described as $N=12$ and $M=3$. MPEG-2 does not insist on a regular GOP structure. For example, a P-picture following a shot-change may be badly predicted since the reference picture for prediction is completely different from the picture being predicted. Thus, it may be beneficial to code it as an I-picture instead.

For a given decoded picture quality, coding using each picture type produces a different number of bits. In a typical example sequence, a coded I-picture was three times larger than a coded P-picture, which was itself 50% larger than a coded B-picture.

4.3.9.3 Buffer control

By removing much of the redundancy from the source images, the coder outputs a variable bit rate. The bit rate depends on the complexity and predictability of the source picture and the effectiveness of the motion-compensated prediction.

For many applications, the bit stream must be carried in a fixed bit rate channel. In these cases, a buffer store is placed between the coder and the channel. The buffer is filled at a variable rate by the coder, and emptied at a constant rate by the channel. To prevent the buffer from under- or overflowing, a feedback mechanism acts to adjust the average coded bit rate as a function of the buffer fullness. For example, the average coded bit rate may be lowered by increasing the degree of quantization applied to the DCT coefficients. This reduces the number of bits generated by the variable-length coding, but increases distortion in the decoded image. The decoder must also have a buffer between the channel and the variable rate input to the decoding process. The size of the buffers in the coder and decoder must be the same.

MPEG-2 defines the maximum decoder (and hence coder) buffer size, although the coder may choose to use only part of this. The delay through the coder and decoder buffer is equal to the buffer size divided by the channel bit rate.

4.4 Proposed Algorithm

Watermarking for MPEG_2 compressed data can be done in spatial, DCT or VLC domain. In the VLC domain watermarking of MPEG-2 compressed data has been proposed by Langelaar et al [127] concerning a system, which is basically a LSB system.

Chun Shein et al [150] proposed watermarking in the VLC domain for MPEG-2 Compressed data in the case of a fragile system. Mobasseri et al [129] also proposed watermarking for a robust system in the VLC domain for MPEG-2 compressed data. However, none of the reported work proposes reversible watermarking in the VLC domain.

Online application of watermarking in the spatial domain becomes cumbersome due to associated high computational complexities involved. On the other hand, Watermarking in the DCT domain needs preprocessing operations such as inverse entropy coding and inverse quantisation [144]. The present work, however, involves watermarking of compressed data in the VLC domain by which the computational complexities involved in the spatial domain and preprocessing operations involved in the DCT domain can be spared with.

Reversibility of watermarking in the same VLC domain, developed the algorithm and tested the same for a two dimensional image in the shape of a single I-frame of a MPEG-2 video sequence.

The algorithm has been discussed under three separate headings:

1. Compression
2. Embedding/ Detecting
3. Reversibility

4.4.1 Compression

The present work adopts MPEG compression method, which is a universally accepted method for data compression. In this work, a shift algorithm further compresses the data that has already been compressed in MPEG2 format. Raw data x is first MPEG2 compressed. For this data is divided into I, P or B frames or fields. The encoded bits that are generated for I frame or I field in the Tuple format are passed through the New Shift Algorithm. The generated bits i.e. C_x are the Compressed bits. The Compression scheme is presented in fig.2 and fig.3.

MPEG compression exploits both the intra and inter redundancy for compression. DCT coding is used to remove intra-frame redundancy and motion compensation is used to remove inter-frame redundancy. Each frame or field is separated into a number of macro

blocks. These macro blocks are 8X8 blocks. Two-dimensional DCT is performed on each of these blocks. After quantization [2], these blocks are tuple coded. The tuple-coded blocks are then VLC coded [3]. The shift algorithm defined below manipulates the tuple-coded run level pairs. This o give rise to new run-level pairs such that they occupy less number of bits. Thus compression is attained.

Shift Algorithm

In the tuple domain, all the levels are right shifted by one bit. The new binary number so formed is represented by its VLC code word. The bit discarded by the right shift is stored in user-defined space provided in MPEG2 standard. This operation results in a net gain of some bits. Here a problem is encountered when a 1 ,2 or 3 is encountered. This is because a 1, 2 or 3 are all stores as a 1. This can be overcome by storing two bits for a 2 or a 3. In some images due to the presence of large number of 1, 2 and 3 as level may not result in a reduced bit stream. In such a condition 1 , 2 or 3 level are left as such. Shifting starts at levels of 4 and 5. To represent 2, 4 and 5, 2 bits are stored in the user defined space.

Steps

The level of the Tuple is right shifted by one bit.

The Bit so discarded is stored in the user-defined space of Mpeg2.

If the level is, one it is stored as such. For levels of one, two, or three two bits are stored in user-defined space. This is because all three are stored as a one in the level. Data bits are stored in the user-defined space according to the following scheme.

For a level of 2 data bits “00” are stored

For a level of 4 data bits “01” are stored

For a level of 5 data bits “10” are stored

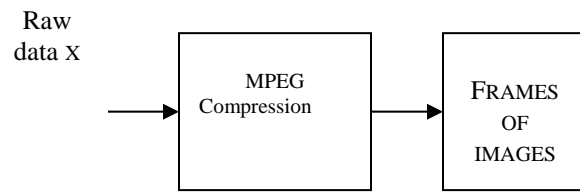


Figure 4.7 Conversion to Frames and fields

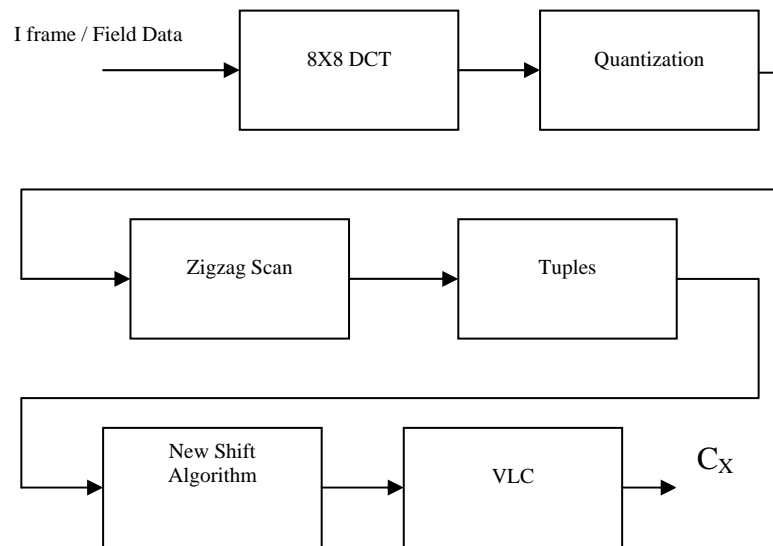


Figure 4.8 Compression scheme

Example:

Let a tuple be (0, 4)

VLC representation is (0 0 0 0 1 0 0 s) 8 bits

$$4_{(10)} = 1\ 0\ 0_{(2)}$$

After right shift it becomes 1 0

$$10_{(2)} = 2_{(10)}$$

The new tuple now is (0,2)

VLC representation is 0 1 0 0 s 5 bits

(Here s is the sign bit)

So there is a net gain of $8 - (5 + 1) = 2$ bits

4.4.2 Embedding / Detecting

The bits embedded (w_i) can be the raw watermark bits or the bits embedded can be encrypted bits for more security.

Embedding Algorithm

- Key K_1 is generated according to the watermark to be embedded. If the embedding bit is a 0 a 1 is generated and if the embedding bit is a 1 a -1 is generated.
- These numbers in K_1 are arranged as a tuple with 0 run. These are then inserted in the main tuple according to another random key K_2 which is in the increasing order.
- Key K_2 is a set of random numbers in the increasing order. The maximum number being the row of the main tuple generated from the original data and the number of such numbers being the size of the watermark to be embedded.

For example:

Let the numbers Key K_1 be [1 -1 1 1 -1]

Key K_2 be [2 3 4 8 10]

Let the main tuple be :

Run	Level
0	3
1	4
2	7
3	2
2	6
6	9
5	7
6	12
4	24
2	9
1	7
0	4

Size of K_1 is 5 so size of K_2 is also 5. Number of rows in the main tuple is 12 so the maximum number that K_2 can have is 12.

K_1 is converted to the run level tuple :

Run	Level
0	1
0	- 1
0	1
0	1
0	- 1

This tuple is inserted in the main tuple according to the key K_2 . Therefore the Embedded tuple now becomes:

Run	Level
0	3
1	4
0	1
2	7
0	-1
3	2
0	1
2	6
6	9
5	7
6	12
0	1
4	24
2	9
0	-1
1	7
0	9

- C bit stream for tuple embedded with the w_i bits is generated according to table B.14 of MPEG-2 standard.

Detecting Algorithm:

- The bit stream is parsed.
- Run level tuple is generated.
- According to key K_2 the positions where key k_1 has been embedded is identified.
- K_1 gives the Watermark.

4.4.3 Reversibility

The reconstructed run level pairs are considered 1 by one and considered for decompression. The number present in the user-defined space is concatenated to the new tuple. This generates the old tuple. The restored tuple is now passed through the existing MPEG2 decoders to get back the raw data. If shifting has started from 1, when 1 is a new

level, if the 1st bit in the user-defined space is a 1 then the old level can be decoded to a 1 and when a zero is encountered in the user defined space then old level is a 2 or a 3 depending on whether the next bit is a 0 or a 1. If shifting has started from 4, when 2 is a new level, if the 1st bit in the user-defined space is a 1 then the old level can be decoded to a 2 and when a zero is encountered in the user defined space then old level is a 4 or a 5 depending on whether the next bit is a 0 or a 1. Under such a condition, when 1 or 3 is encountered no de-shifting is applied.

4.5 Result

Two-dimensional pictures, which can be likened to I-Frame pictures of MPEG-2, have been considered as test cases. Both subjective as well as objective evaluations are carried. The pictures considered are: Lena, Goldhill, Baboon, Barbara and Peppers. The watermark considered is a 64 x 64 logo image shown in Fig.4.9

Subjective evaluation:

The pictures are first Compressed in MPEG-2. The pictures are then compressed in the proposed scheme format. The watermark logo picture is embedded according to the proposed scheme. The original images, images recovered (after decompression of MPEG-2 compressed images) and the reconstructed images (after reversibly extracting the watermark) are shown in Fig.4.10 - Fig. 4.24. The absolute error images with the original images, and reconstructed images as input, are shown in Fig. 4. 25 - Fig. 4.29. The absolute error images with the original images and recovered images as input are shown in Fig.4.30- Fig.4.34 (the absolute value is multiplied by a factor of 5). These two sets of pictures are visually equal showing that the watermark has been extracted reversibly.



Figure 4.9 64X64 Watermark image



Fig.4.10Original Lena



Figure 4.11 PSNR1 = 36.9859(For MPEG-2 Compressed)



Figure 4.12 PSNR2 = 36.9859(Compressed using Proposed Algorithm)

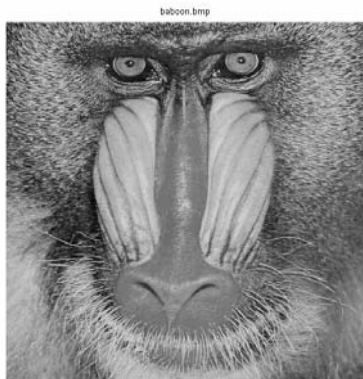


Figure 4.13 Original Baboon

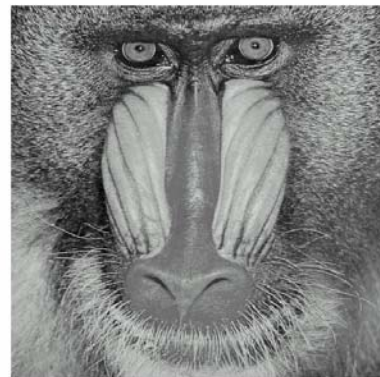


Figure 4.14 PSNR1 = 31.8904(MPEG-2 Compressed)

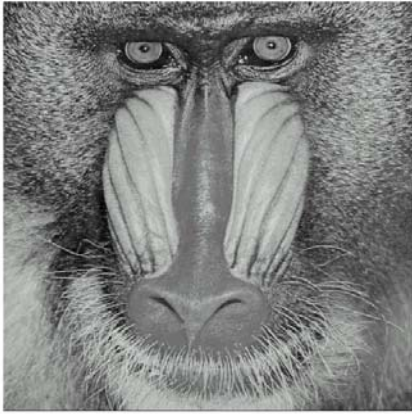


Figure 4.15 PSNR2 = 31.8904 (Compressed using Proposed Algorithm)

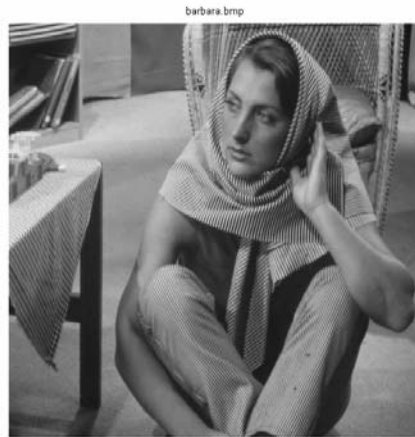


Figure 4.16 Original Barbara



Figure 4.17 PSNR1 = 34.9343(For MPEG-2 Compressed)

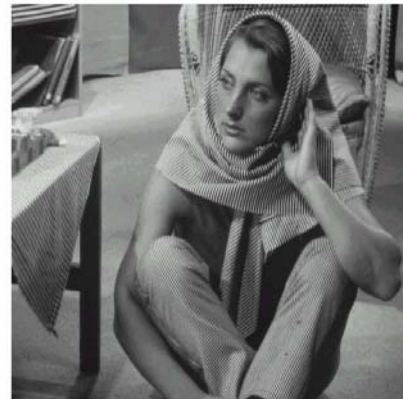


Figure 4.18 PSNR2 = 34.9343(Compressed using Proposed Algorithm)

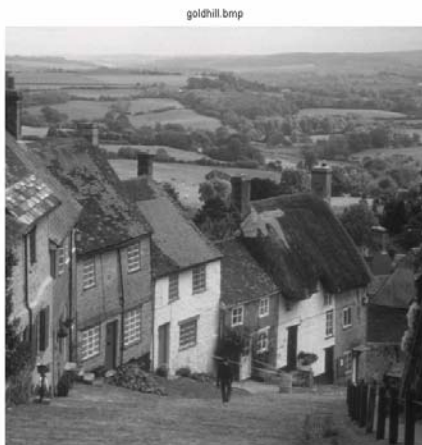


Figure 4.19 Original Goldhill



Figure 4.20 PSNR1 = 34.7470(For MPEG-2 Compressed)



Figure 4.21 PSNR2 = 34.7470(Compressed using Proposed Algorithm)



Figure 4.22 Original Peppers



Figure 4.23 PSNR2 = 34.9343(For MPEG-2 Compressed)



Figure 4.24 PSNR2 = 34.9343(Compressed using Proposed Algorithm)

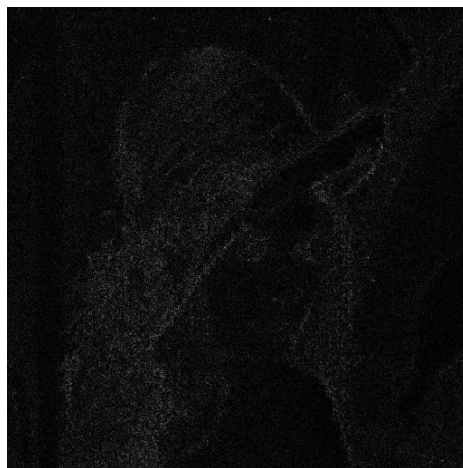


Figure 4.25 Absolute Difference between Original Lena & Recovered Lena



Figure 4.26 Absolute Difference between Original Baboon & Recovered Baboon

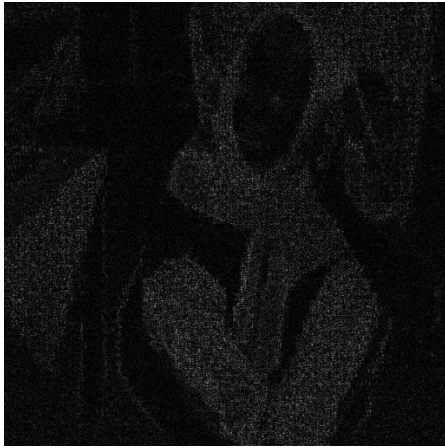


Figure 4.27 Absolute Difference between Original Barbara & Recovered Barbara

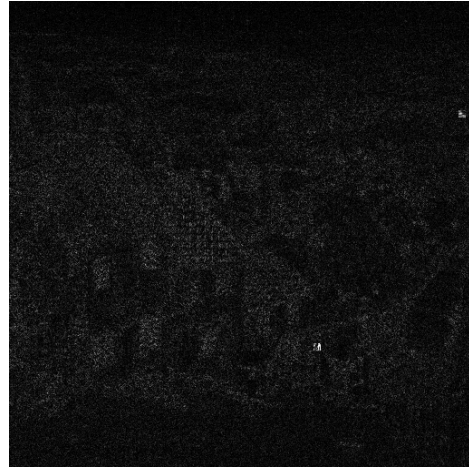


Figure 4.28 Absolute Difference between Original Goldhill & **Recovered** Goldhill

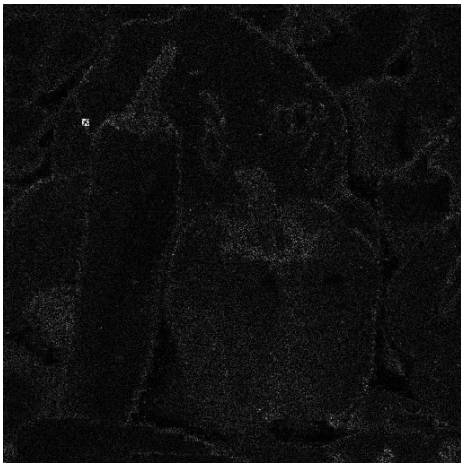


Figure 4.29 Absolute Difference between Original Peppers & Recovered Peppers

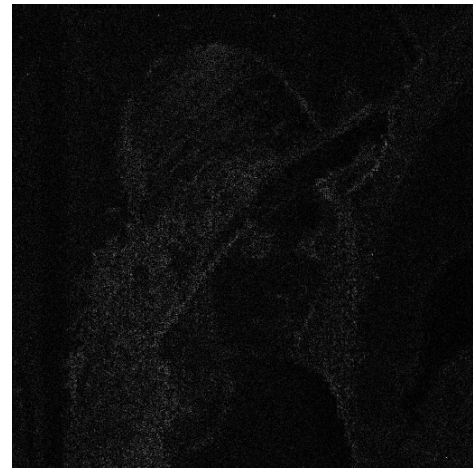


Figure 4.30 Absolute Difference between Original Lena & Reconstructed Lena



Figure 4.31 Absolute Difference between Original Baboon & Reconstructed Baboon

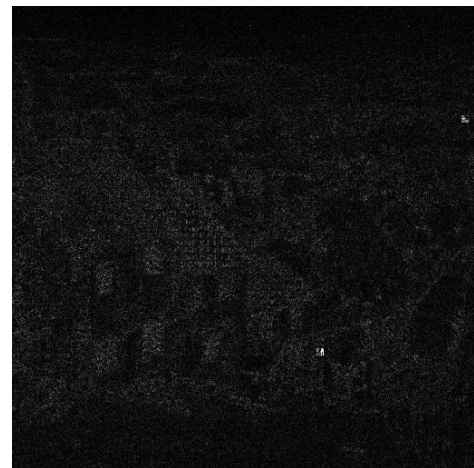


Figure 4.32 Absolute Difference between Original Goldhill & **Reconstructed** Goldhill

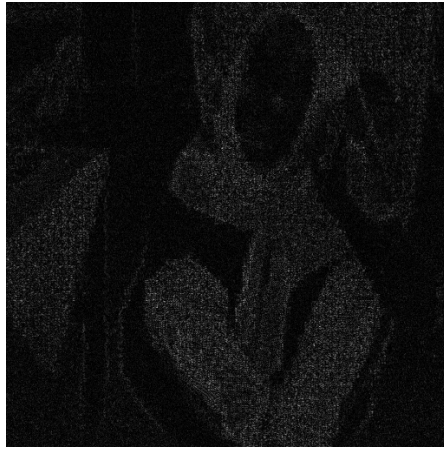


Figure 4.33 Absolute Difference between Original Barbara & Reconstructed Barbara

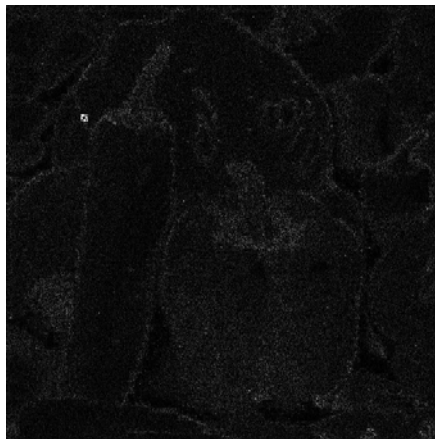


Figure 4.34 Absolute Difference between Original Peppers & Reconstructed Peppers

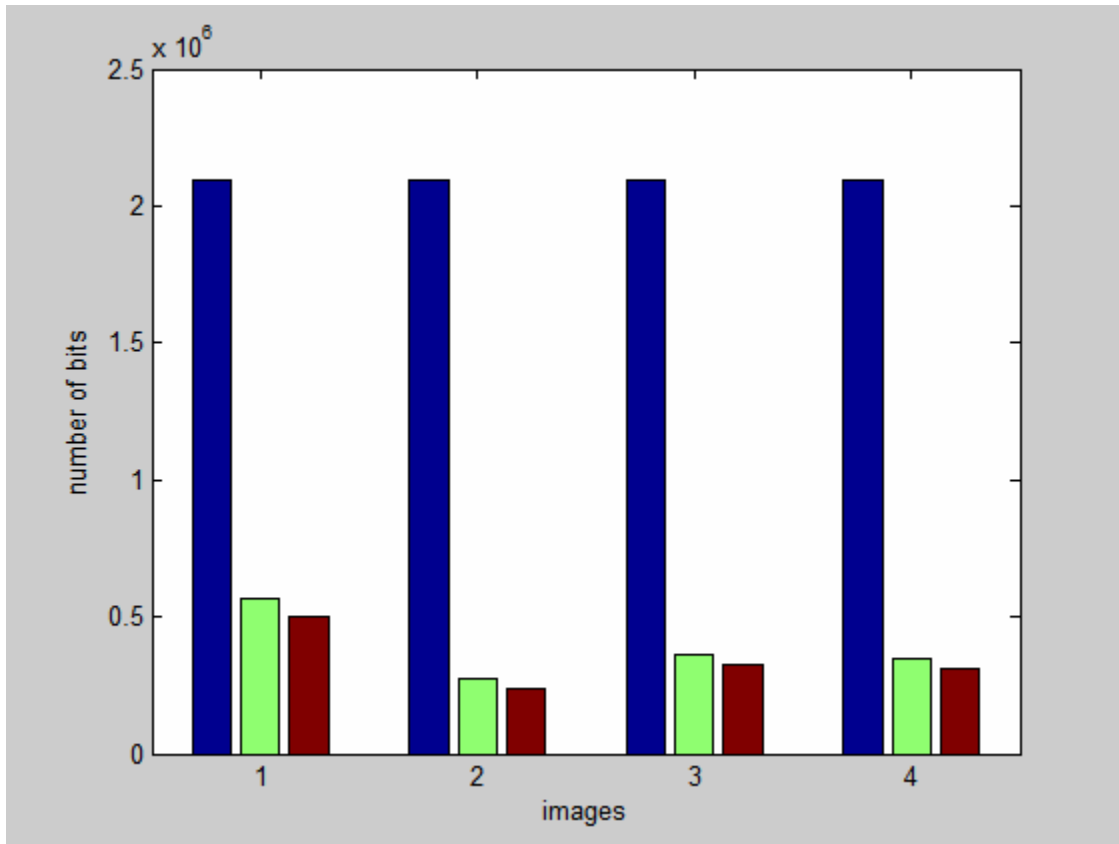


Fig. 4.35 Number of Bits in Raw, MPEG-2 Compressed Data and Proposed Algorithm Compressed Data
(The blue bar represents raw data, the green bar represents MPEG-2 Compressed data and the brown bar represents data compressed in the proposed algorithm)

Objective evaluation:

The PSNR of the recovered (after decompression of MPEG-2 compressed data) and reconstructed (after reversibly extracting the watermark) images are calculated. The number of the bits in the MPEG-2 compressed form is calculated for each picture. The number of bits in the compressed form is calculated for each picture. This is represented in the bar (Fig.4.35). The PSNR and MSE of the recovered and reconstructed images are calculated and presented in Table 4.6..

The PSNR and MSE of the recovered and reconstructed images are identical. This shows that the watermark has been extracted reversibly, and the reconstructed image is equal to the unwatermarked image pixel-by-pixel and bit-by-bit.

Table 4.5 PSNR, MSE and Bit Representation at Different Stages of the Algorithm						
	PSNR1 (MPEG-2 Compressed)	PSNR2 (New - algorithm)	No. of bits in MPEG-2	No. of bits in new algorithm = sum of bits in compressed form and bits in user defined space	MSE	Gain of bits
Lena	36.9859	36.48	270610	173446+63592 =237037	13.0165	33573
Baboon	31.8904	31.8904	568841	445456+58531 =503991	65.6152	64850
Barbara	34.9343	34.9343	360107	287376+34887 = 322263	20.8749	37844
Goldhill	34.7470	34.7470	345824	278284+32752 = 311036	21.7964	34788
Peppers	35.7140	35.7140	288250	232553+23312= 255865	17.4455	32385

4.6 Conclusion

In chapter 4 a reversible watermarking method in the VLC domain of MPEG2 data has been described. By exploring the redundancy in the already compressed MPEG2 data reversibility is achieved. This allows the new algorithm to interoperate with existing MPEG-2 algorithm and encourages the investment needed in integrated circuits to make the technology cheap.

CHAPTER 5

ENERGY CLUSTER BASED WATERMARKING SCHEME USING GA

OBJECTIVE: To propose a new energy cluster based robust watermarking scheme using genetic algorithm.

CHAPTER ORGANISATION:

5.1 Insight into GA

5.2 Requirement of Watermarking Algorithm to prove authenticity

5.3 Issues addressed in the Algorithm

5.4 Proposed Algorithm

5.5 Simulation results

5.6 Conclusion and scope for future work

5.1 Insight into Genetic Algorithm

Genetic algorithms are a part of **evolutionary computing**, which is a rapidly growing area of artificial intelligence. These are inspired by Darwin's theory of evolution. An evolutionary process solves problems, resulting in a best (fittest) solution (survivor) - in other words, the solution is evolved.

A brief history of genetic algorithm

5.1.1 History

Evolutionary computing was introduced in the 1960s by I. **Rechenberg** in his work "*Evolution strategies*" (*Evolutionsstrategie* in original). The idea presented in this book was then developed by other researchers. **Genetic Algorithms** (GAs) were invented by John **Holland** and developed by him and his students and colleagues. This first finds mention in Holland's book "*Adaptation in Natural and Artificial Systems*" published in 1975. In 1992 John **Koza** used genetic algorithm to evolve programs to perform certain tasks. He called his method "**genetic programming**" (GP). LISP programs were used, because programs in this language can be expressed in the form of a "parse tree", which is the object the GA works on.

5.1.2 Chromosome

All living organisms consist of cells. In each cell there is the same set of **chromosomes**. Chromosomes are strings of DNA and serve as a model for the whole organism. A chromosome consists of **genes**, blocks of DNA. Each gene encodes a particular protein. Basically, it can be said that each gene encodes a **trait**, for example color of eyes. Possible settings for a trait (e.g. blue, brown) are called **alleles**. Each gene has its own position in the chromosome. This position is called **locus**. Complete set of genetic material (all chromosomes) is called **genome**. Particular set of genes in genome is called **genotype**. The genotype is with later development after birth base for the organism's **phenotype**, its physical and mental characteristics, such as eye color, intelligence etc.

5.1.3 Reproduction

During reproduction, **recombination** (or **crossover**) first occurs. Genes from parents combine to form a whole new chromosome. The newly created offspring can then be

mutated. **Mutation** means that the elements of DNA are a bit changed. These changes are mainly caused by errors in copying genes from parents. The **fitness** of an organism is measured by success of the organism in its life (survival).

One example of a class of problems, which cannot be solved in the "traditional" way, is NP problems. There are many tasks for which we may apply fast (polynomial) algorithms. There are also some problems that cannot be solved algorithmically. There are many important problems in which it is very difficult to find a solution, but once a solution is reached, it is easy to check the solution. This fact led to **NP-complete problems**. NP stands for non deterministic polynomial and it means that it is possible to "guess" the solution (by some non deterministic algorithm) and then check it. Studying of NP-complete problems is, for simplicity, restricted to the problems where the answer can be a yes or a no. Because there are tasks with complicated outputs, a class of problems called **NP-hard** problems has been introduced. This class is not as limited as class of NP-complete problems. A characteristic of NP-problems is that a simple algorithm, perhaps obvious at a first sight, can be used to find usable solutions. This approach however provides many possible solutions - just trying all possible solutions is very slow process (e.g. $O(2^n)$). For even slightly bigger instances of these types of problems this approach is not usable at all. Today nobody knows if some faster algorithm exists to provide exact answers to NP-problems. The discovery of such algorithms remains a big task for researchers. Today many people think that such algorithms do not exist and so they are looking for an alternative method. An example of an alternative method is the genetic algorithm.

Examples of the NP problems are satisfiability problem, traveling salesman problem or knapsack problem.

Genetic algorithms are inspired by Darwin's theory of evolution. Solution to a problem solved by genetic algorithms uses an evolutionary process (it is evolved). Algorithm begins with a **set of solutions** (represented by **chromosomes**) called **population**. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions that are selected to form new solutions (**offspring**) are selected according to their fitness - the more suitable they are the more chances they have to reproduce.

This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied.

5.1.4 Encoding of a Chromosome

A chromosome should in some way contain information about solution that it represents. The most used way of encoding is a binary string. A chromosome is shown in Fig.4.1.

Table 5.1
Binary string representation of encoded chromosome

Chromosome 1	1101100100110110
Chromosome 2	1101111000011110

Each chromosome is represented by a binary string. Each bit in the string can represent some characteristics of the solution. Another possibility is that the whole string can represent a number. There are many other ways of encoding. The encoding depends mainly on the solved problem. For example, one can encode directly integer or real numbers; sometimes it is useful to encode some permutations and so on.

5.1.5 Crossover

After encoding type has been decided on, the next step is crossover operation. Crossover operates on selected genes from parent chromosomes and creates new offspring. The simplest way how to do that is to choose randomly some crossover point and copy everything before this point from the first parent and then copy everything after the crossover point from the other parent.

Crossover can be illustrated in table 5.2. Here (| is the crossover point)

Table 5.2
Representation of Crossover

Chromosome 1	11011 00100110110
Chromosome 2	11011 11000011110
Offspring 1	11011 11000011110
Offspring 2	11011 00100110110

There are other ways to make crossover, for example more crossover points can be chosen. Crossover can be quite complicated and depends mainly on the encoding of chromosomes. Specific crossover made for a specific problem can improve performance of the genetic algorithm.

5.1.6 Mutation

After a crossover is performed, mutation takes place. Mutation is intended to prevent falling of all solutions in the population into a local optimum of the solved problem. Mutation operation randomly changes the offspring resulted from crossover. In case of binary encoding a few randomly chosen bits can be switched from 1 to 0 or from 0 to 1. Mutation is illustrated in table 5.3.

Table 5.3
Representation of Mutation

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110110

The technique of mutation (as well as crossover) depends mainly on the encoding of chromosomes. For example while encoding permutations; mutation could be performed as an exchange of two genes.

5.1.7 Crossover and Mutation Probability

There are two basic parameters of GA - crossover probability and mutation probability.

5.1.7.1 Crossover probability: Crossover probability is defined as “how often crossover will be performed”. If there is no crossover, offspring are exact copies of parents. If there is crossover, offspring are made from parts of both parent's chromosome. If crossover probability is **100%**, then all offspring are made by crossover. If it is **0%**, whole new generation is made from exact copies of chromosomes from old population (but this does not mean that the whole generation is the same) Crossover is made in hope that new chromosomes will contain good parts of old chromosomes and therefore the new

chromosomes will be better. However, it is good to let some part of old population survive to next generation.

5.1.7.2 Mutation probability: Mutation Probability is defined as “how often parts of chromosome will be mutated”. If there is no mutation, offspring are generated immediately after crossover (or directly copied) without any change. If mutation is performed, one or more parts of a chromosome are changed. If mutation probability is **100%**, whole chromosome is changed, if it is **0%**, nothing is changed. Mutation generally prevents the GA from falling into local extremes. Mutation should not occur very often, because then GA will in fact change to **random search**.

5.1.8 Other Parameters

There are also some other parameters of GA. One another particularly important parameter is population size.

Population size: Population size is defined as “how many chromosomes are in population (in one generation)”. If there are too few chromosomes, GA has few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many chromosomes, GA slows down. Research shows that after some limit (which depends mainly on encoding and the problem) it is not useful to use very large populations because it does not solve the problem faster than moderate sized populations.

Chromosomes are selected from the population to be parents for crossover. The problem is how to select these chromosomes. According to Darwin's theory of evolution the best ones survive to create new offspring. There are many methods in selecting the best chromosomes. Examples are roulette wheel selection, Boltzman selection, tournament selection, rank selection, steady state selection and elitism selection etc.

5.1.9 Outline of the Basic Genetic Algorithm

1. **[Start]** Generate random population of n chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness $f(x)$ of each chromosome x in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete

4. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
5. **[Crossover]** With a crossover probability cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
6. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
7. **[Accepting]** Place new offspring in the new population
8. **[Replace]** Use new generated population for a further run of the algorithm
9. **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population
10. **[Loop]** Go to step 2

5.1.10 Genetic Algorithm Progress

The outline of the Basic GA is very general. There are many parameters and settings that can be implemented differently in various problems.

Factors that is specific to a particular application:

1. The choice of chromosomes and the type of encoding
2. Crossover and Mutation. The crossover is the most important parts of the genetic algorithm. Mainly these two operators influence the performance.
3. Selection of parents for crossover. This can be done in many ways, but the main idea is to select the better parents (best survivors) in the hope that the better parents will produce better offspring. Generating populations from only from two parents may cause the loss of the best chromosome from the last population. This is true, and so **elitism** is often used. This means, that at least one of a generation's best solution is copied without changes to a new population, so the best solution can survive to the succeeding generation.

5.1.11 Chromosome selection methods

5.1.11.1 Roulette Wheel Selection

Parents are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have. Imagine a **roulette wheel** where all the chromosomes in the population are placed. The size of the section in the roulette wheel is proportional to the value of the fitness function of every chromosome - the bigger the value is, the larger the section is (Fig. 5.1)

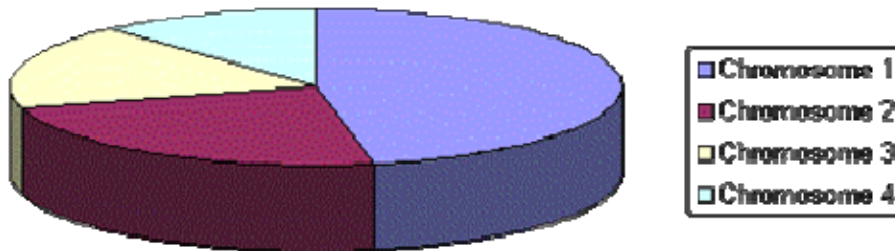


Figure 5.1 Representation of Roulette Wheel Selection

A marble is thrown in the roulette wheel and the chromosome where it stops is selected. Clearly, the chromosomes with bigger fitness value will be selected more times.

Roulette wheel selection algorithm:

1. **[Sum]** Calculate the sum of all chromosome fitness in population - sum S .
2. **[Select]** Generate random number from the interval $(0, S)$ - r .
3. **[Loop]** Go through the population and sum the fitnesses from 0 - sum s . When the sum s is greater than r , stop and return the chromosome where you are.

Step 1 is performed only once for each population.

5.1.11.2 Rank Selection

The previous type of selection will have problems when there are big differences between the fitness values. For example, if the best chromosome fitness is 90% of the sum of all fitness then the other chromosomes will have very few chances to be selected.

Rank selection ranks the population first and then every chromosome receives fitness value determined by this ranking. The worst will have the fitness 1 , the second worst 2 etc. and the best will have fitness N (number of chromosomes in population).

The situation changes after changing fitness to the numbers determined by the ranking (Fig.5.2).

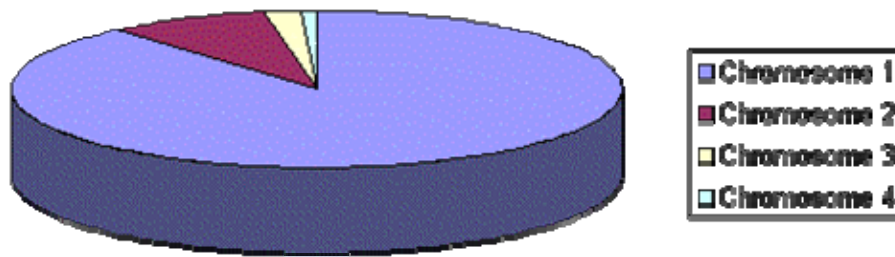


Figure 5.2 a) Situation before Ranking (graph of fitness)

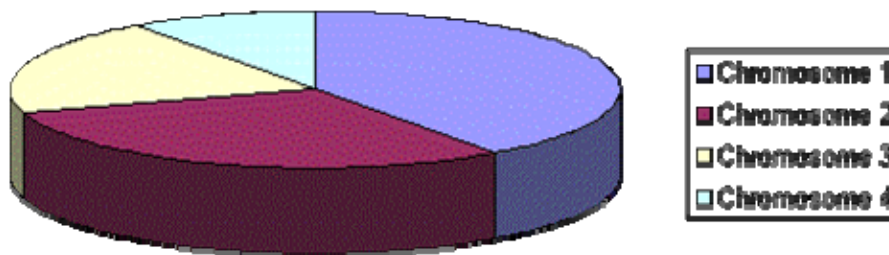


Figure 5.2 b) Situation after Ranking (graph of order numbers)

Now all the chromosomes have a chance to be selected. However this method can lead to slower convergence, because the best chromosomes do not differ so much from other ones.

5.1.11.3 Steady-State Selection

This is not a particular method of selecting parents. The main idea of this type of selecting to the new population is that a big part of chromosomes can survive to next generation.

In steady-state selection GA in every generation a few good (with higher fitness) chromosomes are selected for creating new offspring. Then some bad (with lower fitness) chromosomes are removed and the new offspring is placed in their place. The rest of population survives to new generation.

5.1.11.4 Elitism

While creating a new population by crossover and mutation, there is a big chance, that the best chromosome will be lost.

Elitism is the name of the method that first copies the best chromosome (or few best chromosomes) to the new population. The rest of the population is constructed in ways described above. Elitism can rapidly increase the performance of GA, because it prevents a loss of the best-found solution.

5.1.12 Encoding selection methods

Encoding of chromosomes is the requirement while starting to solve a problem with GA. Encoding depends on the problem heavily.

Some encoding schemes are discussed below:

5.1.12.1 Binary Encoding

Binary encoding is the most common one, mainly because the first research of GA used this type of encoding and because of its relative simplicity.

In **binary encoding**, every chromosome is a string of **bits** - **0** or **1**(Table 5.3)

Table 5.4
Representation of Chromosome with binary encoding

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

Binary encoding gives many possible chromosomes even with a small number of alleles. On the other hand, this encoding is often not natural for many problems and sometimes corrections must be made after crossover and/or mutation.

Example of Problem: Knapsack problem

The problem: There are things with given value and size. The knapsack has given capacity. Select things to maximize the value of things in knapsack, but do not extend knapsack capacity.

Encoding: Each bit says, whether the corresponding thing is in knapsack.

5.1.12.2 Permutation Encoding

Permutation encoding can be used in ordering problems, such as traveling salesman problem or task ordering problem.

In permutation encoding, every chromosome is a string of numbers that represent a position in a sequence (Table 5.5).

Table 5.5
Representation of Chromosome With Permutation Encoding

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Permutation encoding is useful for ordering problems. For some types of crossover and mutation corrections must be made to leave the chromosome consistent (i.e. have real sequence in it) for some problems.

Example of Problem: Traveling salesman problem (TSP)

The problem: There are cities and given distances between them. Traveling salesman has to visit all of them, but he does not want to travel more than necessary. Find a sequence of cities with a minimal traveled distance.

Encoding: Chromosome describes the order of cities, in which the salesman will visit them.

5.1.12.3 Value Encoding

Direct value encoding can be used in problems where some more complicated values such as real numbers are used. Use of binary encoding for this type of problems would be difficult. In the value encoding, every chromosome is a sequence of some values. Values can be anything connected to the problem, such as (real) numbers, chars or any objects as shown in Table 5.6.

Table 5.6
Representation of Chromosome with value encoding

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

Value encoding is a good choice for some special problems. However, for this encoding it is often necessary to develop some new crossover and mutation specific for the problem.

Example of Problem: Finding weights for a neural network

The problem: A neural network is given with defined architecture. Find weights between neurons in the neural network to get the desired output from the network.

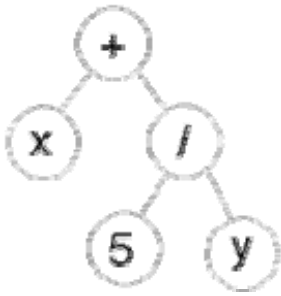
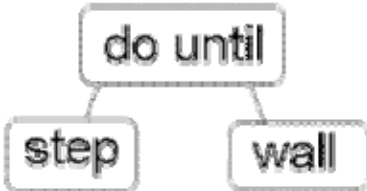
Encoding: Real values in chromosomes represent weights in the neural network.

5.1.12.5 Tree Encoding

Tree encoding is used mainly for evolving programs or expressions, i.e. for **genetic programming**.

In the **tree encoding** every chromosome is a tree of some objects, such as functions or commands in programming language.

Table 5.7
Representation of Chromosome with tree encoding

Chromosome A	Chromosome B
	
$(+ x (/ 5 y))$	$(\text{do until step wall})$

Tree encoding is useful for evolving programs or any other structures that can be encoded in trees.

Example of Problem: Finding a function that would approximate given pairs of values

The Problem: Input and output values are given. The task is to find a function that will give the best outputs (i.e. the closest to the wanted ones) for all inputs.

Encoding: Chromosome are functions represented in a tree.

5.1.13 Crossover and Mutation for Different Kinds of Encoding

Crossover and mutation are two basic operators of GA. Performance of GA depend on them to a large extent. The type and implementation of operators depends on the encoding and also on the problem. There are many ways to perform crossover and mutation. Crossover and mutation for the various encoding schemes are discussed below

5.1.13.1 Binary Encoding

Crossover

Single point crossover - one crossover point is selected, binary string from the beginning of the chromosome to the crossover point is copied from the first parent, and the rest is copied from the other parent



Figure 5.3 a) Single Point Crossover

11001011+11011111 = 11001111

Two point crossover – When two crossover points are selected, binary string from the beginning of the chromosome to the first crossover point is copied from the first parent, the part from the first to the second crossover point is copied from the other parent and the rest is copied from the first parent again



Figure 5.3 b) Two Point Crossover

11001011 + 11011111 = 11011111

Uniform crossover - bits are randomly copied from the first or from the second parent



Figure 5.3 c) Uniform Crossover

$$11001011 + 11011101 = 11011111$$

Arithmetic crossover - some arithmetic operation is performed to make a new offspring



Figure 5.3 d) Arithmetic Crossover

$$11001011 + 11011111 = 11001001 \text{ (AND)}$$

Mutation

Bit inversion - selected bits are inverted



Figure 5.4 Bit Inversion Mutation

$$11001001 \Rightarrow 10001001$$

5.1.13.2 Permutation Encoding

Crossover

Single point crossover - one crossover point is selected, the permutation is copied from the first parent till the crossover point, then the other parent is scanned and if the number is not yet in the offspring, it is added

$$(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9) + (4\ 5\ 3\ 6\ 8\ 9\ 7\ 2\ 1) = (1\ 2\ 3\ 4\ 5\ 6\ 8\ 9\ 7)$$

Mutation

Order changing - two numbers are selected and exchanged

(1 2 3 4 5 6 **8** 9 7) => (1 **8** 3 4 5 6 **2** 9 7)

5.1.13.3 Value Encoding

Crossover

All crossovers from binary encoding can be used

Mutation

Adding a small number (for real value encoding) - a small number is added to (or subtracted from) selected values

(1.29 5.68 **2.86** **4.11** 5.55) => (1.29 5.68 **2.73** **4.22** 5.55)

5.1.13.4 Tree Encoding

Crossover

Tree crossover - one crossover point is selected in both parents, parents are divided in that point and the parts below crossover points are exchanged to produce new offspring

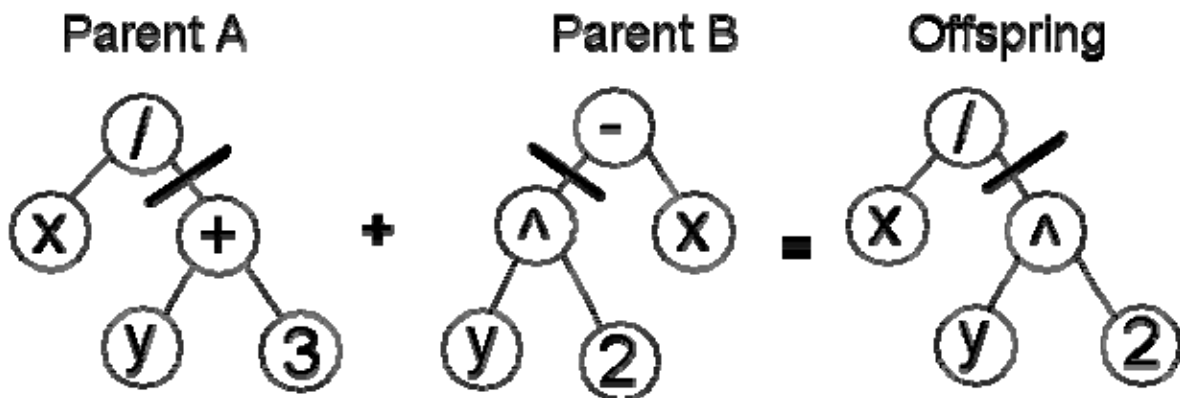


Figure 5.5 Tree Encoding Crossover

Mutation

Changing operator, number - selected nodes are changed

5.1.14 Parameters of GA

Crossover rate

Crossover rate should be high generally, about **80%-95%**. (However some results show that for some problems crossover rate about 60% is the best.)

Mutation rate

Mutation rate should be very low. Best rates seem to be about **0.5%-1%**. High mutation rate would lead random calculation.

Population size

Very big population size usually does not improve performance of GA (in the sense of speed of finding solution). Good population size is about **20-30**, however sometimes sizes 50-100 are reported as the best. Research in this area has shown that the best population size depends on the **size of encoded string** (chromosomes). It means that the population should be higher for chromosomes with 32 bits, than for chromosomes with 16 bits.

Selection

Any of the selection types can be used. In this work **Elitism** selection has been used.

Encoding

Encoding **depends on the problem** and also on the size of instance of the problem. Here Permutation encoding has been used

5.1.15 Applications of GA

Genetic algorithms have been used for difficult problems (such as NP-hard problems), for machine learning and also for evolving simple programs. They have been also used for some art, for evolving pictures and music.

In this work GA has been used to reach at an adequate trade off between imperceptibility and robustness of a watermarking scheme.

The advantage of GA is in its parallelism. GA is traveling in a search space using more individuals (and with genotype rather than phenotype) so that it is less likely to get stuck in a local extreme like the other methods. It is also easy to implement. Once the basic GA algorithm has been implemented, it can be used to solve another problem by writing a new chromosome (just one object), with the same encoding only the fitness function need be changed. However, for some problems, choosing and implementation of encoding and fitness function can be difficult.

The disadvantage of GA is in its computational time. GA can be slower than other methods. In this work GA is applied to clusters of the image instead of the whole image to decrease the computation time.

5.2 Requirement of Watermarking Algorithm to Prove Authenticity

The Watermarking addresses a wide range of issues. The requisites of a particular scheme are determined by its application. For establishing ownership, or determining authenticity, the watermarking scheme should satisfy certain issues [2]

- 1) High fidelity or Unobtrusiveness: the watermark should be imperceptible i.e. the watermarked image should be perceptually equal to the original image. The watermark should not degrade or affect the image quality
- 2) Robust: the watermark should be resistant to attacks both intentional and unintentional

Specifically the following type of attacks:

- I. Geometric distortions: Operation such as cropping, scaling, rotation and translation.
 - II. Collusion attacks: Attempts to destroy the watermark by making use of multiple watermarked images. Also it should not be possible to generate a new watermarked image by combining several watermarked images.
 - III. Other operations: This includes digital to analog conversion, analog to digital conversion, resampling, requantization and compression distortion (e.g. JPEG or MPEG)
- 3) Large data capacity: the data hiding capacity should be large so that more secret information can be embedded in the image

5.3 Issues addressed in the Algorithm

This work attempts to obtain an adequate trade off between the first two characteristics using GA. Optimization result given by GA is good for solving this constrained problem [151], i.e. it gives high fidelity and good robustness. The processing speed is however very low. It might take days to embed a particular image [152]. The current paper attempts to address this problem by applying clusters of the image [153] instead of the full image as a whole. The clusters in [153] are ART2 based. In this paper the clusters are formed according to their energy distribution, which is DCT, based [154,155] but unlike it this method does not use feature extraction points. This not only reduces the requirement of a large number of images during the training phase of the generation of

clusters for a particular image, but also reduces computational complexity and gives higher values of fidelity and Robustness.

5.4 Proposed Algorithm

5.4.1 Watermark Embedding Algorithm

The watermark Embedding Algorithm is divided into two Phases.

5.4.1.1 Training Phase

This results in an optimization Table OT. The OT provided coefficients where embedding can be done to obtain higher PSNR values (higher fidelity) and higher NCC values (better robustness to attacks)

The Training Algorithm is as depicted below:

1. 8X8 DCT of the image is obtained
2. The 8X8 blocks are scanned in a zigzag fashion
3. Energy is calculated for each block is defined in (5.1)

$$i = \sum_{1}^{64} D(i)^2 \quad \text{Eqn.5.1}$$

Here $D(i)$ is the value of the i^{th} DCT coefficient of the block

These blocks are arranged according to their decreasing energy, in the process scrambling the DCT blocks. This gives rise to clusters. Number of such blocks is defined in (5.2)

$$\frac{I_x \cdot I_y}{64} = N \quad \text{Eqn.5.2}$$

I_y and I_y are image dimensions. If 'a' were the number of clusters and each cluster has 'b' 8X8 blocks, then number of such clusters is defined by 'n' in (5.3)

$$n = \frac{N}{b} \quad \text{Eqn.5.3}$$

4. A reference table is calculated for each cluster, which is the ratio between DC and AC coefficients as shown in (5.4).

$$R_n(i) = \sum \frac{y_k(i)}{y_k(1)} \quad i = 2, 3, \dots, 64. \quad \text{Eqn.5.4}$$

$$n = 1, 2, \dots, b$$

n and b are as in step 4. So the reference table has n sets of R -values

5. A set P is defined which defines a relation of the DC values of one block and the current AC coefficient for embedding.

$$P_k(i) = 1 \quad \text{If } y_k(i) \cdot R_n(i) \leq y_k(0) \quad \text{Eqn.5.5}$$

$$P_k(i) = 0 \quad \text{Otherwise.}$$

Watermarked pixel value

$$y'_k(i) = y_k(i) \quad \text{if } P_k(i) = w(i) \quad \text{Eqn.5.6}$$

$$y'_k(i) = -1 \cdot y_k(i) \quad \text{if } P_k(i) \neq w(i) \quad \text{Eqn.5.7}$$

6. For recovery of the watermark Rn' is calculated for each cluster and the watermark bit extracted in accordance with (5.8) and (5.9)

$$w' = 1 \quad \text{if } y_{k''}(i) \cdot R'_n(i) \leq y_{k''}(0) \quad \text{Eqn.5.8}$$

$$w' = 0 \quad \text{Otherwise} \quad \text{Eqn.5.9}$$

where $y_{k''}$ is DC coefficient of K^{th} block of the cluster and $y_{k''}$ is the i^{th} AC coefficient..

7. GA is used to obtain coefficients where embedding of the watermark is done so that better PSNR for the watermarked image and NCC for the watermark can be obtained.

$$Fitness = PSNR + \lambda j \cdot \sum_1^j (NCj) \quad \text{Eqn.5.10}$$

5.4.1.2 The GA is as follows

1. 64 random numbers between 2 and 64 are generated.
These are the chromosomes. The magnitude of the chromosome decides in which coefficient of a block of the cluster the watermark will be embedded. For example if the coefficient cluster is 2, 3, 4, then for the 1st block of the cluster the watermark bit will be embedded in the 2nd coefficient. For the 2nd block the watermark bit will be embedded in the 3rd coefficient and so on
2. The cluster is first converted back to its pixel domain. After embedding the new cluster formed is compared to the original cluster and the PSNR is calculated.
3. The new cluster so formed is then passed through mean filter and compressed using MPEG compression. The NCC values are calculated for the recovered watermark image in accordance with equations (5.8) and (5.9).
4. Fitness is evaluated as defined in (5.10)

5. The best-fit coefficients are stored in OT.

Step 7 is evaluated for all the clusters and the best-fit coefficients are stored in OT.

5.4.1.3 Embedding Phase

This results in the watermarked image. The embedding algorithm is defined below.

The Watermarking of the image is done while still in the clustered condition .The watermark bits are embedded in accordance with coefficients from the OT Blocks are now arranged as they were in the original image resulting in the watermarked image.

5.4.2 Watermark Detection Algorithm

For extraction of watermark, the test image is transformed into its DCT domain. The estimated reference table Rn' is produced as described in equation (5.4). Watermark is detected in accordance with equations (5.8) and (5.9).

5.5 Simulation results

The 512X512 Lena image is taken as the input is shown in (Fig.5.3). Populations consisting of 100 chromosomes were evaluated. The embedded watermark of size 64X64 is shown in Fig 5.4.Each bit of the watermark was embedded into one 8X8 non-overlapping block.

The watermarked version of the Lena image generated with the algorithm described in the paper is shown in figure 3. The PSNR obtained is 39.072. Another watermarked version of Lena generated by embedding the watermark into random DCT coefficients is shown in figure 5. for comparison. The PSNR obtained is 37.671.Low pass filtering and MPEG compression attacks these two watermarked versions of Lena, separately. Figure 5 shows the extracted watermarks embedded with the proposed algorithm Figure 6 shows the extracted watermarks embedded into random DCT coefficients.



Figure 5.6. 512X512 Original Lena Image



Figure 5.7. 64X64 Watermark image



Figure 5.8 The Watermarked image generated by Proposed algorithm PSNR 39.671



Figure 5.9 The Watermarked image generated by Embedding into random pixels PSNR 37.671



Figure 5.10 a) Watermark recovered from proposed algorithm NC = 0.9431



Figure 5.10 b) For the proposed algorithm Watermark recovered after mean filtering of the watermarked image NC = 0.5388



Figure 5.10(c) For the proposed algorithm Watermark recovered after compressing the watermarked image NC= 0.5372



Figure 5.11 a) For the proposed algorithm Watermark recovered after mean filtering of the watermarked image. Watermark cannot be detected



Figure 5.11 b) For the proposed algorithm Watermark recovered after compressing the watermarked image .Watermark cannot be detected

5.7 Conclusion and scope for future work

An energy cluster based watermarking algorithm-using GA is proposed in Chapter 5. Adequate trade off between fidelity and Robustness has been obtained. GA has been applied to clusters of the image instead of the complete image so the processing speed is higher.

Further research work can be carried to increase data hiding capacity, using Multiple Optimization GA.

CHAPTER 6

CONCLUSION

Conclusion

- Existing Watermarking algorithms in the following domains were studied.

1. Spatial
2. Transform
3. Compressed

In contrast to the spatial-domain-based watermarking, frequency-domain-based techniques can embed more bits of watermark and are more robust to attack. Online application of watermarking for video in the spatial domain becomes cumbersome due to associated high computational complexities involved. On the other hand, Watermarking in the DCT domain needs preprocessing operations such as inverse entropy coding and inverse quantization. Watermarking of compressed data in the VLC domain, the computational complexities involved in the spatial domain and preprocessing operations involved in the DCT domain can be spared with.

- Watermarking algorithms have varied requirements according to the application, the algorithm aims to target. Two such requirements have been dealt with in the thesis. They are:
 1. Reversibility: A reversible watermarking scheme has been proposed in the VLC domain of MPEG-2 compressed data. Simulation results show that PSNR of MPEG-2 compressed data is equal to the PSNR of the recovered data.
 2. Trade off between imperceptibility and Roustness. Simulation results show that adequate tradeoff between imperceptibility and robustness has been achieved using cluster based genetic algorithm.

Bibliography

1. W.Diffie and M.E.Hellman, “New Directions in Cryptography”, IEEE trans. on Information Theory, Vol.IT-22, No.6, Nov.1976.
2. B.M.Macq, J.J.Quisquater, “Cryptography for Digital TV Broadcasting”, Proc. of the IEEE, Vol.83, No.6, pp 944-957 , Jun1995.
3. G.J.Simmons, “The History of Subliminal Channels”, IEEE Jou. On selected areas in Communications, Vol.16, No.4, pp.452-462 , May1998.
4. G.J.Simmons, “Results Concerning the Bandwidth of Subliminal Channels”, IEEE Jou. on selected areas in Communications, Vol.16, No.4, pp.463-473, May1998.
5. Bumster et.al., “A Progress Report on Subliminal-free Channels”, Proc. of First Int. Workshop on Information Hiding, Cambridge, UK, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.), May30-Jun1, 1996.
6. Neal Koblitz, A Course in Number Theory and Cryptography, Springer-Verlag.
7. C. Meadow and I.S.Maskowintz, “Covert Channels – A content based view”, Proc. of First Int. Workshop on Information Hiding, Cambridge, UK, Lecture notes in Computer Science, Vol.1174, Ross Anderson (Ed.), May30-June1 1996.
8. M. Acken, “How Watermarking Value to Digital Content”,Comm. of ACM, Vol 41, No.7, pp 75-77, July 1998.
9. E. Franz, et. al., “Computer Based Steganography”, Proc. First Intl. Workshop on Information Hiding, Cambridge, UK, May 30 – June 1, 1996
10. Homer, The Iliad (trans. R. Fragels), Middlesex, England: Penguin 1972.
11. Herodotus, The Histories (trans. R. Selincourt), Middlesex, England: Penguin 1972.
12. David Kahn, “The History of Steganography”, Proc. of First Int. Workshop on Information Hiding, Cambridge, UK, May30-June1 1996.
13. Mitchell D.Swanson et all, “Multimedia data embedding and Watermarking technologies ” Proceedings of the IEEE, Vol 86(6) pp. 1064-1087, June 1998

14. Hal Berghel, "Watermarking Cyberspace", Comm. of the ACM, Vol.40, No.11, pp.19-24, Nov.1997.
15. N.R.Wagner, "Finger Printing", Proc. of the 1983 Symposium on Security and Privacy, 1983, Oakland, California, IEEE Computer Society, pp.18-22, Apr.25-27.
16. Adrian G. Bor_s and Ioannis Pitas "Image watermarking using block site selection and DCT domain constraints" (C) OSA OPTICS EXPRESS, Vol. 3, No. 12, pp 512- 523, Nov 1998
17. S.Craver et al., "Can Invisible Watermarks Resolve Rightful Ownership?", IBM Research Report, RC205209, Jul25, 1996. <http://www.research.ibm.com/>
18. Seolc Kang and Yoshinao Aoki "Digital Image Watermarking by Fresnel Transform and its Robustness". IEEE journal, pp- 221- 225, 1999.
19. Saraju P Mohanty, N. Ranganathan and Ravi K. Namballa , "VLSI Implementation of Invisible Digital Watermarking Algorithms Towards the Development of a Secure JPEG Encoder", IEEE journal, pp- 183 – 188 ,2003.
20. Graham Shaw, "Digital Document Integrity " , ACM Multimedia Workshop Marina Del Rey CA USA, pp- 143- 144, 2000.
21. Chia-Mu Yu and Chun-Shien Lu, "Robust Non-Interactive Zero-Knowledge Watermarking Scheme Against Cheating Prover", ACM Portal MM-SEC'05, New York, New York, USA ACM , pp 103 – 110, August 1–2, 2005.
22. Liu Yongliang , Wen Gao, "Secure Watermark Verification Scheme", IEEE International Conference on Multimedia and Expo (ICME), pp - 923- 926, 2004.
23. S.Baudry, J.F.Delaigle, B.Sankur, B.Macq, and H.Maitre, "Analyses of error correction strategies for typical communication channels in watermarking," Signal Process, Vol 81, No.6,pp.1239-1250,June 2001.
24. International Federation of the Phonographic Industry, "Request for proposals", Embedded Signaling Systems Issue 1.0. 54 Regent Street, London W1R 5PJ, June 1997.

25. Fabien A.P. Petitcolas and Ross J. Anderson, "Weaknesses of copyright marking systems", Multimedia and security workshop at ACM Multimedia '98. Bristol, UK, September 1998.
26. Jiri Fridrich et al "Comparing robustness of watermarking techniques", Electronic imaging 1999, The International Society for Optical Engineering, Security and Watermarking of Multimedia Contents , Vol 3657, San Jose , CA, USA, January 1999
27. M.Kutter, F.A.P. Petitcolas, "A fair benchmark for image watermarking systems", Electronic Imaging '99. Security and Watermarking of Multimedia Contents, vol.3657, The International Society for Optical Engineering, San Jose, CA, USA, January 1999
28. I.Pitas and T. Kaskalis, "Applying Signature on Digital Images" , Proc IEEE, Workshop on Non-linear Signal and Image Processing, I. Pitas(Ed.), pp. 460-463,1995
29. G.Voyatzis, N . Nikolaides , I. Pitas, "Digital Watermarking: An Overview", Proceedings of IX European Signal Processing Conference (EUSIPCO), pp. 13-16,Island of Rhodes, Greece September 1998.
30. Ross J. Anderson , Fabien A.P. Petitcolas, "On the limits of Steganography. IEEE Journal of Selected Areas in Communications", 16(4) pp. 474-481, May 1998
31. M.A. Suhail and M.S. Obaidat, "Digital Watermarking-based DCT and JPEG model", IEEE Trans. On Instrumentation and Measurement , Vol 52, No. 5, October 2003
32. I.J. Cox and M.L.Miller , "A review of watermarking and the importance of perceptual modeling", in the Proceedings of SPIE : Storage and Retrieval for Image and Video Databases V , San Jose , CA, 1997.
33. G.C Langelaar , "Conditional access to television service" Wireless Communication, the interactive multimedia CD-Rom, 3rd edition ,Baltzer Science Publishers Amsterdam, 1999.

34. G.C Langelaar I Setyawan and R.L.Lagendijk, "Watermarking image and video data : A state-of-the-art overview " IEEE Signal Proc. Magazine, Vol.17, No.5, pp. 20-46, September 2000.
35. T.Kalker, G.Depovere, J .Haitsma and M. Maes, "A video watermarking system for broadcast monitoring", in the Proceedings of SPIE Secirity and Watermarking of Multimedia Contents, San Jose,CA, 1999.
36. A.M. Alattar, Smart images using Digimarc's watermarking technology, in the proceedings of SPIE, Security and Watermarking of Multimedia Contents II, San Jose , Ca, 2000.
37. M Chen, Y. He and R.L. Lagendijk, "Error Detection by fragile Watermarking "in the Proceedings of the 22nd Picture Coding Symposium PCS 2001, Seoul, Korea April 2001.
38. P. Campisi, D Kundur , D. Hatzinakos and A . Neri, "Hiding-based Compression for Iproved Color Image Coding" in the Proc. Of SPIE: Security and Watermarking of multimedia Contents IV, Vol. 4675, pp. 230-239, San Jose, CA, 2002
39. Y Hwang, B. Jeon, "Error detection in a compressed video using fragile watermarking " in the Proceedings IEEE, ICME 2002, Vol.I, pp 129-132, August 2002
40. Campisi, G. Giunta, A.Neri, "Object-based Quality of Service Assessment using Semi-fragile Tracing Watermarking in MPEG4 Video Cellular Devices" in Proceedings of IEEE, ICIP, Vol. II, pp 881-884, Rochester, NY, 2002
41. J. Zhao and E. Koch, "Embedding Robust Labels into Images for Copyright Protection", Proc of International Congress on Intellectual Property Rights for Specialized Information Knowledge and New Technologies, Vienna, Austria, , pp 242-25,1 August 21-25 1995
42. Cox et al., "Secure Spread Spectrum Watermarking for Multimedia," /EEE Trans. Image. Proc., Vol. 6, No 12, pp. 1673-87, December. 1997.
43. B. Chen and G. W. Wornell, "Dither Modulation: A New Approach to Digital Watermarking and Information Embedding," Security and Watermarking of Multimedia Contents, Proc SPIE. Vol 3657, San Jose, CA, January. 1999.

44. J. J. Eggers, J.K. Su, and B. Girod, "A Blind Watermarking Scheme Based on Structured Codebooks," IEE , Secure Images and Image Authentication. London, UK, April 2000.
45. S. Voloshynovskiy et al.. "Attack Modeling: Towards a Second Generation Watermarking Benchmark," Sig. Processing. Special Issue on Information Theoretic Issues in Digital Watermarking, Vol. 81, No. 6, pp. 1177-214, 2001.
46. F. Hartung. J. K. Su. and B. Girod. "Spread Spectrum Watermarking: Malicious Attacks and Counter-Attacks," Security and Watermarking of Multimedia Contents, Proc. SPIE. vol. 3657, San Jose, CA, Jan. 1999.
47. Craver et al., "On the Invertibility of Invisible Watermarking Techniques," Proc. IEEE Int'l. Conf, Image Processing 7997, vol. 1, pp. 540-543.
48. M. Kutter. S. Voloshynovskiy, and A. Herrigel, "Watermark Copy Attack," IS&T/ SPIE's 72th Annual Symp , Electronic Imaging 2000: Security and Watermarking of Multimedia Content I, P. W. Wong and E. J. Delp. Eds. SPIE Proc., Vol. 3971, San Jose, CA, pp. 371- 380, January 2000.
49. S. Voloshynovskiy et al, "Generalized Watermark Attack Based on Watermark Estimation and Perceptual Remodulation, " IS& T/SPIE'S 12th Annual Symp, Electronic Imaging 2000: Security and Watermarking of Multimedia Content /I, P. W. Wong and E. J. Delp, Eds., SPIE Proc., vol. 3971, San Jose, CA, pp. 358-70, January 2000.
50. J. K. Su and B. Girod, "Power-Spectrum Condition for Energy-Efficient Watermarking," Proc. IEEE ICIP '99, October 1999.
51. J. K. Su, J. J. Eggers, and B. Girod, "Analysts of Digital Watermarks Subjected to Optimum Linear Filtering and Additive Noise," Sig. Processing, Special Issue on Information Theoretic Issues in Digital Watermarking, Vol. 81, No. 6, pp. 1141-75, 2001.
52. S. Voloshynovskiy et al., "A Stochastic Approach to Content Adaptive Digital Image Watermarking," Int'l. Wksp.Info. Hiding, Vol. LNCS 1768, Lecture Notes in Comp. Sci., Springer Verlag, pp. 212-36 ,29 September. -1 October. 1999.

53. Sviatolsav Voloshynovskiy, Shelby Pereira, Thierry Pun, University, "Attacks on Digital Watermarks: Classification, Estimation-Based Attacks, and Benchmarks" IEEE Communications Magazine August 2001.
54. Mauro Barni, Franco Bartolini "Watermarking Systems Engineering—Enabling Digital Assets Security and Other Applications" by Marcel Dekker, Inc. 2004.
55. K. Hirotsugu, "An Image Digital Signature System with ZKIP for the Graph Isomorphism", Proc IEEE, International Conf. On Image Processing, Vol-3, pp 247-250, 1996.
56. C-T Hsu, J-LWu "Hidden digital watermarks in images," IEEE Trans. Image Processing, vol. 8, pp. 58–68, January 1999.
57. C. I. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models," IEEE J. Select. Areas Commun., Vol. 16, pp. 525–539, May 1998.
58. Z. H. Wei, P. Qin, and Y. Q. Fu, "Percept digital watermark of images using wavelet transform," IEEE Trans. Consumer electron, Vol. 44, pp. 1267–1272, November 1998.
59. R. Dugad, K. Ratakonda, and N. Ahuja, "A new wavelet-base for watermarking image," in Proc. Int. Conf. Image Processing, Vol. 2, pp. 419–423, 1998.
60. C. -T. Hsu and J. -L. Wu, "Multiresolution watermarking for digital images," IEEE Trans. Consumer Electron., Vol. 45, pp. 1097–101, August 1998.
61. Y.-S. Kim, O.-H. Kwon, and R.-H. Park, "Wavelet based watermarking method for digital images using human visual system," Electron. Lett., Vol. 35, pp. 466–468, March 1999.
62. H.-J. Wang and C.-C. J. Kuo, "Image protection via watermarking on perceptually significant wavelet coefficient," in Proc. IEEE 2nd Workshop Multimedia Signal Processing, pp. 279–284, December 1998.
63. Lumini and D. Maio, "A wavelet-based image watermarking scheme," in Proc. Int. Conf. Information Technology, pp.122–126, 2000
64. J. R. Hernandez, M. Amado, and F. Perez-Gonzalez, "DCT-domain watermarking techniques for still image: Detector performance analysis and a new structure," IEEE Trans. Image Processing, Vol. 9, pp. 55–68, January 2000.

65. M.-J. Tsai, K.-Y. Yu, and Y.-Z. Chen, "Joint wavelet and spatial transformation for digital watermarking," *IEEE Trans. Consumer Electron.*, Vol. 46, pp. 137–144, February 2000.
66. H.-J. Wang and C.-C. J. Kuo, "An integrated progressive image coding and watermark system," in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing*, Vol. 6, pp. 3721–3724, May 1998.
67. L. Xie and G. R. Arce, "Joint wavelet compression and authentication watermarking," in *Proc. IEEE Int. Conf. Image Processing*, Vol. 2, pp. 427–431, October 1998.
68. J.J.K O'Ruanidh et. al. "Watermarking Digital Images for Copyright Protection", *IEE Proc. Vision Image and Signal Processing*, Vol 143, No 4, August 1996
69. J.J.K. O'Ruanidh et. al. "Phase Watermarking on Digital Images", *Proc. IEEE, International Conf. On Image Processing, ICIP-96*, Vol 3, pp 239-242,1996
70. M.D.Swanson et. al., "Transparent Robust Image Watermarking", *Proc IEEE, International Conf. On Image Procesing*, Vol 3, pp 211-214, 1996.
71. I.J.Cox et. al., "Secure spread Spectrum Watermarking of Images,Audio and Video", *Proc IEEE 1996 International Conf on Image Processing*, Vol 3, pp 243-246 http://www.neci.nj.nec.com/tr/neci_tr_95_10.ps, 1996.
72. A.G.Bors and I. Pitas, "Image Watermarking using Image Domain Constraints", *Proc IEEE 1996 International Conf. On Image Processing*, Vol 3, pp 231-234, 1996.
73. C.Podilchuk and W.Zeng, "Perceptual Watermarking of Still Images", 1997 IEEE, First Workshop on Multimedia signal Processing, Priceton, New Jersey, USA, pp 363-368, June 23-25 1997
74. A.Piva et. al., "DCT based Watermark Recovery without resorting to the Uncorrupted Original Signal", *IEEE 1997 International Conf. on Image Processing*, Vol.1, pp 520- 523,1997.
75. J.J.K O'Ruanidh and T. Pun , "Rotation , Scale and Translation Invariant Digital Image Watermarking", *Proc IEEE 1997, International Conf. On Image Processing* ,Vol 1,pp 536-539,1997

76. J.R.Smith and B.O.Comiskey, "Modulation and Information Hiding in Images",
Proc of First International Workshop on Information Hiding, University of
Canbridge, UK, Lecture Notes in Comp. Sc., Vol 1174, Ross Anderson (Ed.),
May 30-June 1 1996
77. D.J.Fleet and D.J.Heeger , "Embedding Invisible Information in color Images",
Proc IEEE 1997, International Conf. On Image Processing, Vol 1, pp 532-535,
1997
78. G.W.Barudaway, "Protecting Publicly available Images with Invisible
Watermark", Proc IEEE 1997,International Conf on Image Processing,Vol 1 pp
524-527, 1997
79. C.I.PodilChuk and W.Zeng, "Image Adaptive Watermarking Visual Models",
IEEE Journal on Selected Areas in Communications ,Vol 16, No 4, pp 525-539,
May 1998.
80. M.Barni et al., "A DCT-Domain System for Robust Image Watermarking", Signal
Processing, Vol 66, No 3, pp 357-372 ,May 1998.
81. I.J.Cox et. al., "A Secure Robust Watermarking for Multimedia", Proc of First
International Workshop on Information Hiding, Lecture Notes in Comp. Sc.,
Speinger-Verlag, Vol.1174, pp 185-206, 1996
82. B.Tao and B.Dickinson, "Adaptive Watermarking in DCT Domain", Proc IEEE
1997, International Conf on Accoustics, Speech and Signal Processing, ICASSP-
97, Vol 4, pp 2985-2988,1997
83. B.M.Macq, J.J.Quisquater, "Cryptography for Digital TV Broadcasting", Proc. of
the IEEE, Vol.83, No.6, pp944-957 , Jun1995.
84. 4 R. G. Wolfgang and E. J. Delp, "A Watermark for Digital Images", ICIP-96,
Vol 3, pp 219-222, 1996.
85. Juan R. Hernández,Martín Amado, Fernando Pérez-González, "DCT-Domain
Watermarking Techniques for Still Images: Detector Performance Analysis and a
New Structure" in IEEE Trans on Image Processing, Vol. 9, No. 1, January 2000 .
86. Wu Huang, Yun Q. Shi, Yi Shi, "Embedding Image Watermarks in DC
Components" in IEEE Trans on Circuits and Systems for Video Technology,
Vol.10, No.6, September 2000.

87. Shih-Hsuan, Yang , Hsin-Chang Chen, "Bit-Plane Watermarking for ZeroTree Coded Imageg "in IEEE pp. 73-78 2002
88. Daubechies "Ortogonal bases of compactly supported wavelets", Comm. Pure Appl. Math. Vol XI pp. 909-996, 1988.
89. D. Kundur and D. Hatzinakos, "A robust digital image watermarking method using wavelet based fusion," in International Conference on Image Processing, vol. 111, pp. 544-547, 1997.
90. X.-G. Xia, C. G. Boncelet, and G. R. Arce, "A multiresolution watermark for digital images," in International Conference on Image Processing, Vol. 111, pp. 548-551, 1997.
91. R. Dugad, K. Ratakonda, and N. Ahuja, "A new wavelet-base for watermarking image," in Proc. Int. Conf. Image Processing, Vol. 2, pp. 419-423, 1998
92. H.-J. M.Wang, P.-C. Su, and C.-C. J. Kuo, "Wavelet-based digital image watermarking," Opt. Express, Vol. 3, no. 12, pp. 491-496, December. 7, 1998.
93. C.-T. Hsu and J.-L. Wu, "Multiresolution watermarking for digital images," *IEEE Trans. Circuits Syst. II*, Vol. 45, pp. 1097-1101, August 1998.
94. D. Kundur and D. Hatzinakos, "Digital watermarking using multi-resolution wavelet decomposition," in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, Vol. 5, Seattle, WA, pp. 2969-2972 , May 1998.
95. G. Nicchiotti and E. Ottaviani, "Non-invertible statistical wavelet watermarking," in Proc. EUSIPCO'98, Vol. 4, Rhodes, Greece, pp. 2289-2292, September 8-11, 1998.
96. M. D. Swanson, B. Zhu, and A. H. Tewfik, "Multiresolution scene based video watermarking using perceptual models," *IEEE J. Select. Areas Commun.*, Vol. 16, May 1998.
97. W. Zhu, Z. Xiong, and Y.-Q. Zhang, "Mutliresolution watermarking for images and video," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 9, pp. 545-550, June 1999.
98. H. Inoue, A. Miyazaki, A. Yamamoto, and T. Katsura, "A digital watermark based on the wavelet transform and its robustness on image compression and

- transformation,” IEICE Trans. Fund. Electron., Commun., Comput. Sci., Vol. E82-A, pp. 2–10, Jan. 1999.
99. I. Pitas, “A Method for signature casting on Digital Images”, Proc. IEEE 1996 International Conf. On Image Processing, ICIP-96, Vol 3, pp 215-218
 100. Mauro Barni, Franco Bartolini, Alessandro Piva, “Improved Wavelet-Based Watermarking Through Pixel-Wise Masking ” in IEEE Trans On Image Processing, Vol. 10, No. 5, May 2001
 101. Xiangui Kang, Jiwu Huang, Yun Q. Shi, Yan Lin, “A DWT-DFT Composite Watermarking Scheme Robust to Both Affine Transform and JPEG Compression”, in IEEE Trans. On Circuits and Systems for Video Technology, Vol. 13, No. 8, Aug 2003
 102. R.G. van Schyndel, A. Z. Tirkel, and C. F. Osborne. “A digital watermark” In Proceedings, 1994 IEEE 1st International Conference on Image Processing (ICIP’94) pp 86-90, Los Alamitos, CA, U.S.A Nov 1994
 103. I. Pitas, “A Method for signature casting on Digital Images”, Proc. IEEE 1996 International Conf. On Image Processing, ICIP-96, Vol 3, pp 215-218, 1996.
 104. N. Nikolaidis and I. Pitas, "Copyright Protection of Images Using Robust Digital Signatures", ICASSP-96, Vol. 4, pp 2168-2171, 1996.
 105. G. Voyatzis and I. Pitas, “Application of Total Automorphism in Image Watermarking”, Proc. IEEE 1996, International Conf. On Image Processing, Vol 2, pp 237-240, 1996.
 106. M. Schneider and S. F. Chang, "A Content Based Approach to Image Generation and Authentication", Proc. IEEE 1996 Intl. Conf. Image Processing, ICIP-96, Vol-3, pp 227-230, 1996.
 107. R. B. Wolfgang and E. J. Delp, "A Watermarking Technique for Digital Imagery: Further Studies", Proc. Intl. Conf. on Imaging Sciences, Systems and Tech. Los Vegas. <http://dynamo.ecn.purdue.edu/ac/delp-pub.html>, Jun 30-Jul 3, 1997
 108. Fred Mintzer, Gordon W. Braudaway and Minerva M. Yeung. “Effective and ineffective digital watermarks”, In Proceedings, 1997 IEEE International Conference on Image Processing (ICIP’97), pp- 9-12, Santa Barbara, CA, U.S.A. October 1997.

109. .M.Yeung and F. Mintzer, "An Invisible Watermarking technique for Image Verification", Proc. IEEE 1997, International Conf on Image Processing, Vol. 2, pp 680-683, 1997.
110. M.Kutter et. al., "Digital Signature of color Images using Adaptive Modulation", Proc.SPIE-EI97, 1997, pp 518-526
111. N. Nikolaidis and I. Pitas, " Robust image watermarking in the spatial domain", Signal Processing , 66(3): pp 385-403, May 1998.
112. Juan R. Hernandez, Fernando Perez-Gonzalez, Jose Mauel Rodriguez and Gustavo Nieto. Performance analysis of a 2-d-multipulse amplitude modulation scheme for data hiding and watermarking of still images" in IEEE Journal on Selected Areas in Communications, 16 No.4 : pp 510-524, May 1998.
113. Da-Chun Wu and Wen-Hsiang Tasi, " Image hiding in spatial domain using an image differencing approach", in CVGIP'98, pp 280-287, Taipei, Taiwan, August 1998
114. W. Bendor, et. al., "Techniques for Data Hiding", IBM Systems Jrnl., Vol. 35, No. 3 and 4, pp 313-336.
115. J.F.Delaiglee et. al., "Watermarking Algorithm based on Human Visual Model",Signal Processing, Vol 66, No 3, pp 319-335, May 1998
116. R. Bangaleea and H.C.S. Rughoopth, "Performance improvement of spread Spectrum Spatial Domain Watermarking Scheme Through Diversity and Attack Characterisation" , in IEEE conference Africon , pp 293-298 ,2002
117. Dipti Prasad Mukherjee, Subhamoy Maitra and Scott T. Acton, "Spatial Domain Digital Watermarking of Multimedia Objects for Buyer Authentication", in IEEE transactions on multimedia, Vol 6. No. 1, Feb. 2004
118. R.B.wolfgang, C.I.Podilchuk, E.J.Delp, "The Effect of Matching Watermark and Compression Transform in Compressed Color Images ," IEEE Int. Conf. Image Process, Vol1., pp.440-444,
119. T.L. Wu and S.F. Wu, "Selective encryption and watermarking of MPEG video," in Proc. Int. Conf. Image Science, Systems, and Technology, CISST'97, June 1997.

120. F. Hartung and B. Girod, "Digital watermarking of raw and compressed video," in Proc. SPIE 2952: Digital Compression Technologies and Systems for Video Communication, pp. 205-213, October 1996.
121. F. Hartung and B. Girod, "Watermarking of MPEG-2 encoded video without decoding and re-encoding," in Proc. Multimedia Computing and Networking 1997 (MMCN 97), San Jose, CA, February 1997.
122. F. Hartung and B. Girod, "Digital watermarking of MPEG-2 coded video in the bitstream domain," in Proc. ICASSP 97, Vol. 4, Munich, Germany, pp. 2621-2624, Apr. 21-24, 1997.
123. F. Hartung and B. Girod, "Copyright protection in video delivery networks by watermarking of pre-compressed video," in Multimedia Applications, Services and Techniques—ECMAST'97 (Springer Lecture Notes in Computer Science, Vol. 1242), S. Fdida and M. Morganti, Eds. Heidelberg, Germany: Springer, pp. 423-436, 1997.
124. F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283-301, May 1998. [47] ISO/IEC13818-2:1996(E), Information Technology—Generic Coding of Moving Pictures and Associated Audio Information, Video International Standard, 1996.
125. G.C. Langelaar, "Real-time watermarking techniques for compressed video data," Ph.D. dissertation, Delft University of Technology, The Netherlands, January 2000.
126. A. Hanjalic, G.C. Langelaar, P.M.B. van Roosmalen, J. Biemond, and R.L. Lagendijk, Image and Video Databases: Restoration, Watermarking and Retrieval (Advances in Image Communications, Vol. 8). New York: Elsevier Science, 2000.
127. G.C. Langelaar, R.L. Lagendijk, and J. Biemond, "Real-time labeling of MPEG-2 compressed video," *J. Visual Commun. Image Representation*, Vol. 9, No. 4, pp. 256-270, December 1998.
128. ISO/IEC13818-2:1996(E), Information Technology—Generic Coding of Moving Pictures and Associated Audio Information, Video International Standard, 1996.

129. Bijan G. Mobasseri et al, "Watermarking of MPEG-2 Video in Compressed Domain Using VLC Mapping", ACM Portal, MM-SEC'05, New York, New York, USA, August 1-2, 2005
130. C. Honsinger, P. Jones, M. Rabbani, and J. Stoffel, "Lossless recovery of an original image containing embedded data," US Patent, No. US6278791, 1999.
131. J.Fridrich, M. Goljan, and R. Du, "Invertible authentication," Proc. SPIE, Security and Watermarking of MultimediaContents, San Jose, California, January 23-26, 2001.
132. J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding new paradigm in digital watermarking," EURASIP J. Appl. Signal Processing, Vol.2002, No.2, pp.185-196, February 2002.
133. G. Xuan, J. Zhu, J. Chen, Y. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform," IEE Electronics Letters, Vol.38, No.25, pp.1646-1648, December 2002.
134. Z. Ni, Y. Shi, N. Ansari, and W. Su, "Reversible data hiding," IEEE Proceedings of ISCAS'03, Vol.2, pp.912-915, May 2003.
135. J. Tian, "Reversible watermarking by difference expansion," Proceedings of Workshop on Multimedia and Security, 19- 22, December 2002.
136. B.Yang; M.Schmucker; W.Funk; C.Busch; S.Sun, "Integer DCT-based reversible watermarking for images using companding technique," Proc. SPIE, Security and watermarking of Multimedia Content, Electronic Imaging, San Jose (USA), 2004.
137. B.Yang, M. Schmucker, X. Niu, C. Busch, S.Sun, "Reversible image watermarking by histogram modification for integer DCT coefficients," IEEE Proceedings of Multimedia Signal Processing Workshop, Siena, Italy, September 2004.
138. G.Xuan, C.Yang, Y.Zhen, Y.Shi and Z.Ni, "Reversible data hiding based on wavelet spread spectrum," IEEE Proceedings of Multimedia Signal Processing Workshop, Siena, Italy, September 2004.
139. G.Xuan, C.Yang, Y.Zhen, Y.Shi and Z.Ni, "Reversible data hiding using integer wavelet transform and companding technique," IWDW 2004.

140. M. Thodi and Jeffrey J. Rodríguez, “Reversible watermarking by prediction-error expansion,” 6th IEEE Southwest Symposium on Image Analysis and Interpretation, Lake Tahoe, USA, March 2004.
141. B.Yang; M.Schmucker; X.Niu; C.Busch; S.Sun, “Integer DCT based reversible image watermarking by adaptive coefficient modification,” SPIE-EI, Security and watermarking of Multimedia Content, Electronic Imaging, San Jose (USA), 2005.
142. ISO/IEC 11172: 'Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s'.
143. ISO/IEC 13818: 'Generic coding of moving pictures and associated audio (MPEG-2)'.
144. MPEG2 Standard ,ISO/IEC 1318 – 2: 1996(E)
145. 'Encoding parameters of digital television for studios', CCIR Recommendation 601-1 XVIth Plenary Assembly Dubrovnik, Vol. XI, Part 1, pp. 319-328,1986.
146. JAIN, A.K.: 'Fundamentals of digital image processing' (Prentice Hall, 1989).
147. WELLS, N.D.: 'Component codec standard for high-quality digital television', Electronics & Communication Engineering Journal, **4**, (4), pp. 195-202, August 1992.
148. CARR, M.D.: 'New video coding standard for the 1990s', Electronics & Communication Engineering Journal, **2**, (3), pp. 119-124, June 1990.
149. RAO, K.R. and YIP, P.: 'Discrete cosine transform: algorithms, advantages, applications' (Academic Press, 1990).
150. Chun-Shein et al., “ Real-time frame-dependent video watermarking in VLC domain”, Signal Processing: Image Communication 20 (2005), pp . 624–642
151. C.H Huang, J.L Wu, “A watermark Optimistion Technique based on genetic algorithms,” SPIE Electronic Imaging 2000, San Jose, January 2000.
152. F.Petitcolas, R Anderson, and M. Kuhn, “Information Hiding A survey” Proc-IEEE, Vol 87, No.7, pp-1062-1078, 1999.
153. Y.L. Chang,Y.H.Chen, “Art2 Based Genetic Watermarking” Proc- (AINA 05) , 2005

154. Adrian G. Bor_s , Ioannis Pitas “Image watermarking using block site selection and DCT domain constraints” (C) *OSA Optics Express*, Vol. 3, No. 12, pp 512-523, Nov 1998.
155. C.S Shieh, H.C Huang, F.H.Wang, “Genetic Watermarking based on transform domain techniques” *Pattern Recognition*, Vol.3, No.37, pp 555-556 March 2004.